

České vysoké učení technické v Praze  
Fakulta elektrotechnická

katedra počítačů

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Dominik Hons**

Studijní program: Otevřená informatika  
Obor: Softwarové systémy

Název tématu: **Hodnocení vín - soutěž**

Pokyny pro vypracování:

Pro potřeby aplikace pro hodnocení vín realizujte její soutěžní část, tedy a) konfiguraci soutěže b) průběh soutěže (uvažujte různá možnosti hodnocení a varianty složení komisí) c) hodnocení veřejností. Podle zadavatelových podkladů a dodaných dat zpracujte analýzu use case, doplňte datové modely a proveďte implementaci s pomocí frameworku Django. Následně vše otestujte pomocí vhodných testů (minimálně pomocí Selenium testů).

Očekávané výstupy práce:

1. Analýza use case a návrh datového modelu
2. Implementace ve frameworku django
3. Selenium testy

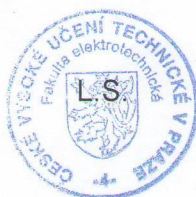
Seznam odborné literatury:

Craig Larman. 2001. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

Vedoucí: Ing. Ondřej Macek, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

doc. Ing. Filip Železný, Ph.D.  
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 14. 4. 2015

České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů



Bakalářská práce

## **Hodnocení vín - soutěž**

*Dominik Hons*

Vedoucí práce: Ing. Macek Ondřej Ph.D.

Studijní program: Otevřená informatika, Bakalářský

Obor: Softwarové systémy

19. května 2015



## Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé práce Ing. Ondřeji Mackovi Ph.D. za všechny rady a připomínky k průběhu prací i k následnému psaní. Dále bych chtěl poděkovat své rodině za podporu.



## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 15. 5. 2015

.....



# Abstract

This work is part of a project whose desire is to create a web application for wine evaluation at wine competitions. The work implements various tools for configuring competitions such as choosing evaluation type or controlling the evaluation process. The competition organizers can easily supervise juries and their members. This work adds a new method of evaluation of wines, a scale of stars. This new evaluation method is integrated to the system without changing the logic of evaluation. Finally, this work implements functionality that allows the general public to evaluate wines via so-called festivals.

# Abstrakt

Tato práce je součástí projektu, který se zabývá vytvořením modulů v aplikaci pro hodnocení vín na vinařských soutěžích. Práce implementuje prostředky pro konfiguraci soutěží, od nastavení způsobu hodnocení po kontrolu průběhu soutěže. Organizátorům umožňuje snadnou správu soutěže, například mohou obsluhovat průběh hodnocení v komisích a měnit jejich složení. Dále tato práce přidává nový způsob hodnocení vín a to pomocí škály hvězdiček. V práci je prezentováno začlenění tohoto způsobu hodnocení tak, aby nebylo nutné měnit logiku soutěže, pro různé typy hodnocení. Nakonec je v této práci implementována možnost hodnocení vín širokou veřejností pomocí takzvaných festivalů.





# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Analýza</b>	<b>3</b>
2.1	Průběh soutěže vín . . . . .	3
2.2	Funkční požadavky . . . . .	5
2.2.1	Konfigurace soutěže . . . . .	5
2.2.2	OIV . . . . .	6
2.2.3	Hvězdičky . . . . .	6
2.2.4	Průběh soutěže . . . . .	6
2.2.4.1	Komisař . . . . .	6
2.2.4.2	Předseda komise . . . . .	6
2.2.4.3	Organizátor . . . . .	7
2.2.5	Hodnocení veřejností . . . . .	7
2.3	Případy užití . . . . .	8
2.3.1	Aktéři . . . . .	8
2.3.2	Specifikace use case . . . . .	8
2.3.3	Konfigurace soutěže . . . . .	8
2.3.3.1	Organizátor . . . . .	9
2.3.4	Průběh soutěže . . . . .	9
2.3.4.1	Komisař . . . . .	10
2.3.4.2	Předseda komise . . . . .	10
2.3.4.3	Organizátor . . . . .	11
2.3.5	Hodnocení veřejností . . . . .	12
2.3.5.1	Uživatel . . . . .	12
2.4	Matice sledovatelnosti požadavků . . . . .	13
<b>3</b>	<b>Použité technologie</b>	<b>15</b>
3.1	Python . . . . .	15
3.2	Django . . . . .	15
3.2.1	Polymorphic . . . . .	16
3.2.2	Captcha . . . . .	17
<b>4</b>	<b>Návrh</b>	<b>19</b>
4.1	Návrh řešení . . . . .	19
4.1.1	Konfigurace soutěže . . . . .	19

4.1.2	Průběh soutěže . . . . .	19
4.1.3	Hodnocení veřejností . . . . .	20
4.2	Datový model . . . . .	21
4.3	Použití návrhových vzorů . . . . .	21
<b>5</b>	<b>Implementace</b>	<b>23</b>
5.1	Modul Competition . . . . .	23
5.1.1	Model balíčku Competition . . . . .	23
5.1.2	Implementovaná funkčnost balíčku Competition . . . . .	24
5.2	Modul Jury . . . . .	24
5.2.1	Implementovaná funkčnost balíčku Jury . . . . .	24
5.3	Modul Oiv . . . . .	27
5.3.1	Udělování medailí . . . . .	27
5.3.2	Modely balíčku Oiv . . . . .	27
5.3.3	Formuláře . . . . .	27
5.3.4	Strategie . . . . .	28
5.4	Modul Organizer . . . . .	28
5.4.1	Implementovaná funkčnost balíčku Organizer . . . . .	29
5.4.2	Formuláře . . . . .	32
5.5	Modul Festival . . . . .	32
5.5.1	Implementovaná funkčnost balíčku Festival . . . . .	32
5.6	Modul Stars . . . . .	33
5.6.1	Modely balíčku Stars . . . . .	33
5.6.2	Formuláře . . . . .	34
5.6.3	Strategie . . . . .	34
5.7	Zhodnocení implementace . . . . .	34
<b>6</b>	<b>Testování</b>	<b>35</b>
6.1	Selenium . . . . .	35
6.1.1	Jury . . . . .	36
6.1.2	Organizer . . . . .	36
6.1.3	Festival . . . . .	37
6.2	Unit . . . . .	37
<b>7</b>	<b>Zhodnocení a budoucnost práce</b>	<b>39</b>
7.1	Shrnutí práce . . . . .	39
7.2	Budoucnost projektu . . . . .	39
<b>8</b>	<b>Závěr</b>	<b>41</b>
<b>A</b>	<b>Modely případů užití</b>	<b>45</b>
A.1	Průběh soutěže . . . . .	45
<b>B</b>	<b>Specifikace use case</b>	<b>47</b>
B.1	Konfigurace soutěže . . . . .	47
B.2	Průběh soutěže . . . . .	48
B.3	Hodnocení veřejností . . . . .	52

<b>C Implementace</b>	<b>55</b>
C.1 Struktura projektu . . . . .	55
C.2 Model balíčku Competition . . . . .	55
<b>D Testování</b>	<b>57</b>
D.1 Jury . . . . .	57
D.2 Organizer . . . . .	58
D.3 Festival . . . . .	58
<b>E Obsah přiloženého CD</b>	<b>59</b>



# Seznam obrázků

2.1	Proces hodnocení vzorku s využitím vznikající aplikace . . . . .	4
2.2	Aktéři use case . . . . .	9
2.3	Use case model - Konfigurace soutěže . . . . .	10
2.4	Use case model - Hodnocení veřejností . . . . .	12
3.1	Model tříd hodnocení. . . . .	16
3.2	Ukázka vygenerované captcha. Uživatel musí opsat písmena z obrázku. . . . .	17
4.1	Model balíčků aplikace . . . . .	20
4.2	Konceptuální model tříd . . . . .	22
5.1	Tlačítka pro zavření vzorku . . . . .	26
5.2	Stavový diagram vzorku . . . . .	26
5.3	Hodnotící formulář OIV . . . . .	28
5.4	Formulář nastavení soutěže, záložka nastavení. . . . .	29
5.5	Formulář editace komise. . . . .	30
5.6	Přehled soutěže. . . . .	32
5.7	Úvodní strana festivalu. . . . .	33
A.1	Use case model - Průběh soutěže . . . . .	45
C.1	Model tříd balíčku Competition . . . . .	56
D.1	Přehled testů balíčku „Jury“. . . . .	57
D.2	Přehled testů balíčku „Organizer“. . . . .	58
D.3	Přehled testů balíčku „Festival“. . . . .	58



# Seznam tabulek

2.1	Use case 06 Hodnotit vzorek - podrobná specifikace . . . . .	11
2.2	Matrice sledovatelnosti požadavků. V řádcích jsou uvedena čísla požadavků, ve sloupcích čísla případů užití. . . . .	13
6.1	Tabulka případů užití a testovacích případů balíčku „Jury“. . . . .	36
6.2	Tabulka případů užití a testovacích případů balíčku „Organizer“. . . . .	37
6.3	Tabulka případů užití a testovacích případů balíčku „Festival“. . . . .	38





# Kapitola 1

## Úvod

Pod pojmem hodnocení vín se, kromě ochutnávání vín, skrývá mnoho organizační a papírové práce. Soutěže vín probíhají tak, že každý komisař obdrží několik papírových formulářů a do nich vyplňuje své hodnocení. Po skončení soutěže jsou tyto formuláře vyhodnocovány ručně a zdlouhavě a až poté je možné vyhlásit vítěze.

Tento projekt má za cíl sjednotit tyto veškeré procedury, potřebné pro hodnocení vín, do jedné aplikace. Pro potřeby organizátorů má aplikace umožňovat snadné zakládání soutěží, to obnáší vyplnění potřebných nastavení, pozvání komisařů, výběr vín a další. Dále má poskytovat organizátorům přehled probíhajících soutěží, možnost editovat komise, komisaře a vzorky, podívat se na průběžné výsledky, udělit medaile a ocenění. Komisaři a jejich předsedové mají mít pro hodnocení k dispozici přehledný hodnotící formulář s vysvětlivkami jednotlivých atributů<sup>1</sup> a mohou porovnávat hodnocení s ostatními komisaři.

Kromě této soutěžní části je v plánu mnohem víc. Aplikace je určená pro širokou veřejnost, každý si může založit vlastní profil, volně hodnotit a přidávat vína a zakládat soutěže pro své přátele. Kromě soutěží se dají vytvořit také festivaly a výstavy. Festival je volně přístupná soutěž, ve které nejsou žádné komise ani komisaři, vína hodnotí veřejnost. Výstava je spojení několika soutěží a festivalů, například pro několikadenní akce. Dalším požadavkem práce je možnost více typů hodnocení. Oficiální hodnotící formulář obsahuje 8 až 10 položek, podle typu vína, které se hodnotí číselnou škálou [9]. Hodnocení je navrženo tak, že nejlepší možné skóre je 100 a nejhorší 40, dále je možné vzorek vyřadit, pokud vážně porušuje požadavky. Dalším typem hodnocení je hodnocení pomocí hvězdiček. Ideální uplatnění tohoto hodnocení je pro veřejnost na festivalech, protože není potřeba znát a hodnotit několik prvků vína, ale pouze udělit celkový počet hvězdiček.

Na začátku projektu již existoval základní prototyp aplikace. Umožňoval import soutěže ze souboru, hodnocení pomocí OIV stupnice a několik editačních a organizačních prvků. Mým úkolem je přepracovat soutěžní část aplikace, tj. doplnit možnosti konfigurace soutěží a jejich komisí. Nastavení soutěže je potřeba rozšířit o možnosti pro změnu typu hodnocení, udělování medailí a další nastavení. Editací komisí se zjednoduší řešení nestandardních situací během soutěže, například se umožní přidání nových komisařů do komise, změna předsedy, výměna pořadí vzorků a jiné. Dalším úkolem je zobecnit průběh soutěže tak, aby bylo možné

---

<sup>1</sup>Při hodnocení vzorku vína má komisař k dispozici vlastnosti vína podle nastavení soutěže. Například obsah cukru, odrůda, ročník...

použít různé typy hodnocení, například pomocí hvězdiček. V neposlední řadě mám na starost hodnocení veřejností. Toto hodnocení má probíhat na takzvaných festivalech, kde nehodnotí komisaři, ale vzorky může ohodnotit kdokoli.

Jedním z důvodů výběru této práce byl fakt, že se jedná o projekt složený z několika částí, na kterých pracují další studenti. Potřeba spolupráce a propojení jednotlivých částí mezi sebou pro mne znamenala hlavně zodpovědnost a nutnost dokončovat své úkoly včas a kvalitně. To, že na výsledku mé práce závisí ostatní, mě dovedlo ke zlepšení pracovní morálky, efektivity a také kvality. Dalším faktorem výběru je rozsah projektu. V průběhu studia je velikost seminárních prací a jiných projektů limitována délkou semestru. Tento projekt má velké ambice a jeho zaměření je široké.

# Kapitola 2

## Analýza

Tato kapitola se zabývá analýzou a modelováním požadavků a případů užití. Požadovaná funkčnost vychází z podkladů dodaných vedoucím práce, především pak z OIV standardu [9]. V první části kapitoly bych chtěl vysvětlit, jak probíhá soutěž v hodnocení vín. Další část kapitoly je věnovaná funkčním požadavkům a modelováním případů užití a jejich specifikací. V poslední části se pak tyto dvě metody propojí pomocí matice sledovatelnosti požadavků. Chtěl bych podotknout, že mým úkolem nebyl grafický vzhled aplikace, ale pouze poskytnout back end.

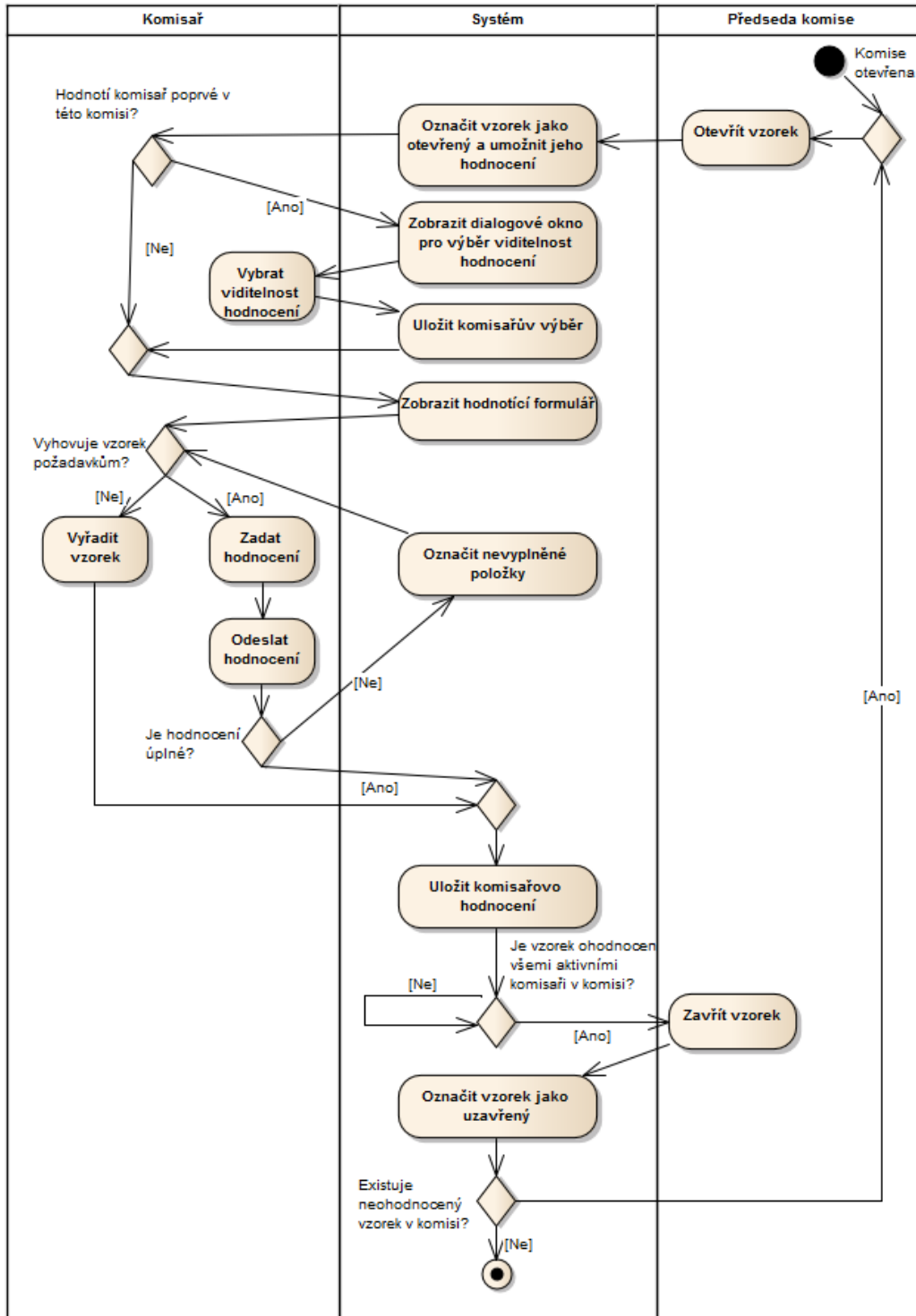
### 2.1 Průběh soutěže vín

Pro lepší pochopení projektu je dobré znát, jak probíhá soutěž vín. Na soutěž jsou předem přihlášeny vzorky vín a jejich kvalita je vyhodnocována kvalifikovanou komisí, ve které by mělo být alespoň pět komisařů. Oficiální podmínky popsané v OIV standardu [9] definují formální náležitosti jako je zacházení se vzorky a chování komisařů. Stupnice hodnocení OIV se dělí do tří tabulek podle typu vína. TW označuje nešumivá vína (anglicky still wines), SW označuje šumivá vína (sparkling and pearl wines) a SB označuje ostatní vinné nápoje (spirituous beverages of vitivinicultural origin). Hodnocení těchto vín se liší v několika attributech a v jejich bodové škále, například šumivost (effervescence) se hodnotí pouze u šumivých vín.

Následně bych chtěl popsat průběh soutěže, který vychází z OIV, ale je modifikován pro potřeby této aplikace. Pro průběh soutěže jsou důležité tři uživatelské role: komisař, předseda komise a organizátor.

Komisař má za úkol hodnotit vzorky a může zvolit, zda tato hodnocení mají být veřejná. Celá komise hodnotí pouze jeden vzorek najednou a proto komisař musí čekat na otevření dalšího vzorku. Komisař také může vzorek vyřadit, pokud zásadně nevyhovuje podmínkám. Kromě toho si může komisař zobrazit porovnání svých předchozích hodnocení. Diagram na obrázku 2.1 modeluje hodnocení jednoho vzorku od jeho otevření až po jeho uzavření. Diagram vychází z kontextu vznikající aplikace.

Předseda komise je komisař, to znamená, že hodnotí vzorky a může dělat vše co komisař, ale má navíc několik funkcí. Zaprvé otevírá a zavírá vzorky, to určuje, který vzorek



Obrázek 2.1: Proces hodnocení vzorku s využitím vznikající aplikace

se aktuálně hodnotí a které vzorky už jsou ohodnocené všemi komisaři. Dále může umožnit komisařům a tedy i sobě změnit své hodnocení aktuálního vzorku. Předseda komise má také k dispozici různé informace o průběhu komise, například počet již uzavřených vzorků, hodnocení komisařů, průměrná hodnocení a další.

Organizátor zodpovídá za správu soutěží, které organizuje. Spouští a uzavírá soutěže a jejich komise, určuje pořadí vzorků v komisích. Také má mnoho možností jak editovat komise, může změnit předsedu komise, přidávat, odstraňovat a deaktivovat komisaře a editovat jejich údaje. Organizátor má možnost ručně přidávat ocenění a medaile vzorkům. Nakonec také určuje, zda se mají zveřejnit výsledky soutěže.

Kromě soutěže se v aplikaci vyskytuje událost nazvaná festival. Festival je verze soutěže určená pro hodnocení veřejností. Od soutěže se liší tím, že zde nejsou komise a vzorky jsou hodnoceny libovolnými registrovanými uživateli. Dále na festivalu není možné vyřazovat vzorky.

## 2.2 Funkční požadavky

Požadavky jsou základem všech systémů. Jsou v podstatě vyjádřením toho, co by měl systém dělat. Požadavky by měly být jedinečným vyjádřením toho, co by měl systém dělat, nikoli toho, jak by to měl systém dělat. To je nesmírně důležitý rozdíl. Můžeme určit, co by měl systém dělat a jaké chování by měl poskytovat, aniž bychom cokoli říkali o způsobu, jak bude dané funkce dosaženo. Funkční požadavek popisuje požadovanou funkci systému. Oproti tomu nefunkční požadavky formulují omezení kladená na systém nebo proces vývoje.

Pro snazší orientaci a efektivitu jsou následující požadavky uspořádány do taxonomie. Je to hierarchie typů požadavků, kterou lze využít ke kategorizaci. Hlavním důvodem pro užívání typů požadavků je uspořádání většího počtu požadavků do menších a snáze zvládnutelných oborů [1].

### 2.2.1 Konfigurace soutěže

Pojem konfigurace soutěže zahrnuje možnosti nastavení soutěže, které volí její organizátor. Mezi tato nastavení patří například změna logo soutěže nebo změna typu hodnocení vzorků na soutěži.

- req01:** Systém bude organizátorovi umožňovat změnu typu hodnocení soutěže (OIV stupnice, hvězdičky).
- req02:** Systém bude organizátorovi umožňovat změnu loga soutěže.
- req03:** Systém bude organizátorovi umožňovat výběr viditelných atributů vín z jejich seznamu (např. jméno výrobce, objem, obsah cukru).
- req04:** Systém bude organizátorovi umožňovat nastavení udělování medailí. Medaile se mohou udělovat přes všechny vzorky, podle kategorií nebo se medaile neudělují a je možné udělovat medaili pouze jednomu vítězi viz [9].

### 2.2.2 OIV

Tato sekce obsahuje požadavky spojené s hodnocením typu OIV popsaným na začátku této kapitoly v sekci o průběhu soutěží 2.1. Je to typ hodnocení používaný na oficiálních soutěžích a je definovaný ve standardu OIV viz [9].

**req06:** Systém bude umožňovat hodnocení vzorku pomocí OIV stupnice.

### 2.2.3 Hvězdičky

Tato část je obdobná části OIV. Pojmem hvězdičky se myslí typ hodnocení vzorků pomocí škály hvězdiček.

**req05:** Systém bude organizátorovi umožňovat nastavení počtu hvězdiček pro hodnocení v rámci soutěže.

**req07:** Systém bude umožňovat hodnocení vzorku pomocí hvězdiček.

### 2.2.4 Průběh soutěže

Sekce průběh soutěže se zabývá požadavky na systém během soutěže. Sekce je rozdělená do tří kategorií, podle uživatelských rolí.

#### 2.2.4.1 Komisař

**req08:** Systém bude komisaři umožňovat hodnotit vzorek.

**req09:** Systém bude komisaři umožňovat vyřazení vzorku, pokud vzorek vážně porušuje požadavky.

**req10:** Systém bude komisaři umožňovat zobrazit porovnání předchozích hodnocení v soutěži. V porovnání budou všechna komisařova hodnocení v dané soutěži, seřazená podle času hodnocení.

**req11:** Systém bude komisaři zobrazovat přehled povinných atributů vzorku, které jsou organizátorem zveřejněny.

#### 2.2.4.2 Předseda komise

**req12:** Systém bude předsedovi komise umožňovat povolení změny hodnocení vzorku. Pokud některý komisař potřebuje změnit své hodnocení.

**req13:** Systém bude předsedovi komise umožňovat zobrazení přehledu komise. V přehledu bude kód vzorku, stav vzorku, hodnocení jednotlivých komisařů a průměrné hodnocení.

**req14:** Systém bude předsedovi komise umožňovat otevření vzorku, pokud žádný vzorek není otevřen a existuje vzorek, který nebyl hodnocen.

**req15:** Systém bude předsedovi komise umožňovat zavření aktuálně hodnoceného vzorku. Zavřít vzorek lze pouze pokud byl ohodnocen všemi aktivními komisaři v dané komisi. Zavřený vzorek není možné hodnotit.

### 2.2.4.3 Organizátor

**req16:** Systém bude organizátorovi umožňovat spuštění soutěže.

**req17:** Systém bude organizátorovi umožňovat uzavření soutěže. V momentě uzavření soutěže se zavřou i všechny komise v soutěži.

**req18:** Systém bude organizátorovi umožňovat otevření komise.

**req19:** Systém bude organizátorovi umožňovat uzavření komise. V uzavřené komisi není možné hodnotit.

**req20:** Systém bude organizátorovi umožňovat změnu předsedy komise.

**req21:** Systém bude organizátorovi umožňovat deaktivaci komisaře v komise. Deaktivovaný komisař nemůže hodnotit.

**req22:** Systém bude organizátorovi umožňovat aktivaci komisaře v komise.

**req23:** Systém bude organizátorovi umožňovat odebrání komisaře z komise.

**req24:** Systém bude organizátorovi umožňovat přidání komisaře do komise. Bude možné přidat již existujícího komisaře z jiné komise, nebo vytvořit nového.

**req25:** Systém bude organizátorovi umožňovat editaci komisaře v komisi. Editovat půjde jméno, příjmení a heslo.

**req26:** Systém bude organizátorovi umožňovat zobrazení přehledu celé soutěže.

### 2.2.5 Hodnocení veřejností

V této části jsou požadavky pro hodnocení veřejností. Veřejností se myslí registrovaný uživatel, který ale není komisařem na soutěži. Místo toho může hodnotit vzorky na takzvaných festivalech. Co je to festival je vysvětleno v sekci o průběhu soutěží zde 2.1.

**req27:** Systém bude registrovanému uživateli umožňovat hodnocení na probíhajícím festivalu.

**req28:** Systém bude registrovanému uživateli umožňovat zobrazení přehledu všech vzorků na festivalu. V přehledu bude kód vzorku, popis vína, uživatelské hodnocení a odkaz na hodnotící formulář.

**req29:** Systém bude registrovanému uživateli umožňovat porovnání svého hodnocení s hodnocením ověřených komisařů.

**req30:** Systém bude registrovanému uživateli umožňovat změnu svého hodnocení, pokud se rozhodne je změnit.



## 2.3 Případy užití

Modelování případů užití (anglicky use case) je další způsob získávání a dokumentování požadavků. Případy užití vyjadřují, kdo a jakým způsobem bude systém využívat. Model případu užití obsahuje čtyři komponenty [1]:

**Hranice systému:** Ohraničení zobrazené kolem případů užití, jež je vyznačením území nebo hranic modelovaného systému.

**Aktéři:** Jsou to role, přidělené osobám nebo předmětům používajícím daný systém.

**Případy užití:** Činnosti, které mohou aktéři se systémem vykonávat.

**Relace:** Smysluplné vztahy mezi aktéry a případy užití.

### 2.3.1 Aktéři

Pro účely modelování případů užití částí systému, které pokrývá tato práce, je potřeba definovat čtyři kategorie uživatelů:

**Uživatel:** Registrovaný a přihlášený uživatel aplikace.

**Organizátor:** Může spravovat soutěže, editovat jejich komise, komisaře a vzorky.

**Komisař:** Hodnotí vzorky v komisi, do které je přiřazen.

**Předseda komise:** Spravuje průběh hodnocení své komise.

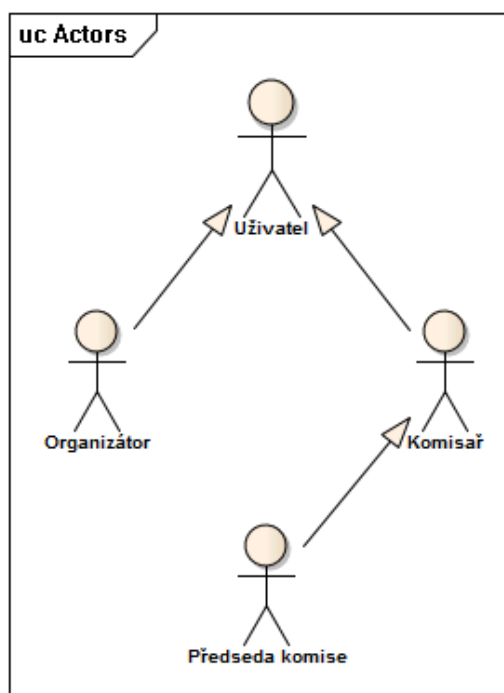
Struktura a vztahy těchto aktérů jsou zobrazeny na obrázku 2.2. Vztahy mezi aktéry označují generalizaci a dědění uživatelských práv. Například komisař je uživatel, který navíc hodnotí na soutěži. Předseda komise je komisař, který spravuje svou komisi, ale zároveň v ní také hodnotí.

### 2.3.2 Specifikace use case

Následující specifikace případu užití obsahují název, jedinečný identifikátor a scénář případu užití. Pro zjednodušení výslovně neuvádím aktéry, kteří často vyplývají ze scénáře. Vstupní podmínky uvádím pouze tam, kde byly použity. Pokud během implementace došlo k nějakým změnám oproti specifikaci, uvádím tuto skutečnost v poznámce. Případy užití jsou rozděleny do tří částí (konfigurace soutěže, průběh soutěže, hodnocení veřejností) a každá tato část je rozdělena podle uživatelských rolí.

### 2.3.3 Konfigurace soutěže

Diagram případu užití je uveden na obrázku 2.3. Podrobné specifikace případů užití jsou uvedeny v příloze B.1.



Obrázek 2.2: Aktéři use case

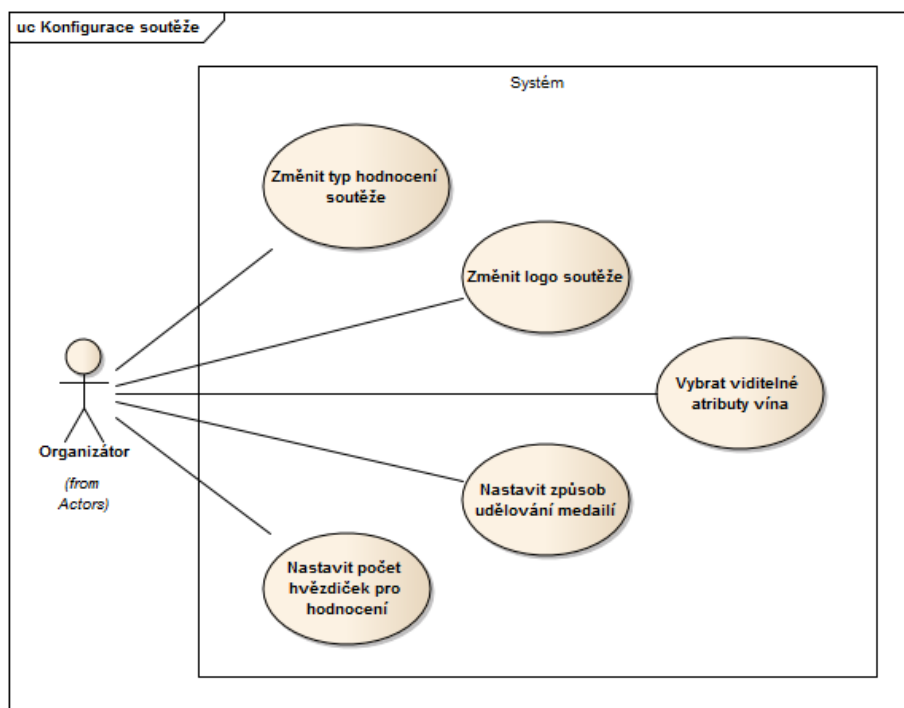
### 2.3.3.1 Organizátor

Následující výčet obsahuje případy užití vztahující se ke konfiguraci soutěže. Popis uživatelské role organizátor je uveden v sekci o průběhu soutěže 2.1.

- UC 01 - Změnit typ hodnocení soutěže
- UC 02 - Změnit logo soutěže
- UC 03 - Vybrat viditelné atributy vína
- UC 04 - Nastavit způsob udělování medailí
- UC 05 - Nastavit počet hvězdiček pro hodnocení

### 2.3.4 Průběh soutěže

Diagram případu užití je rozsáhlejší a proto je uveden v příloze A.1 stejně jako podrobné specifikace případů užití v příloze B.2. Na ukázkou uvádím, v tabulce 2.1, podrobně jeden případ užití, konkrétně use case číslo 06 Hodnotit vzorek.



Obrázek 2.3: Use case model - Konfigurace soutěže

#### 2.3.4.1 Komisař

Následující případy užití jsou definované pro uživatelskou roli komisař. Týkají se hodnocení vzorku na soutěži. Podrobnější popis uživatelských rolí je uveden v sekci o průběhu soutěže 2.1.

- UC 06 - Hodnotit vzorek
- UC 07 - Vyřadit vzorek
- UC 08 - Porovnat své hodnocení

#### 2.3.4.2 Předseda komise

Výčet případů užití zahrnuje možnosti a povinnosti předsedy komise. Ten zodpovídá za průběh hodnocení v komisi.

- UC 09 - Zavřít vzorek
- UC 10 - Otevřít vzorek
- UC 11 - Zobrazit přehled komise
- UC 12 - Povolit změnu hodnocení

Případ užití: Hodnotit vzorek
ID: 06
Stručný popis: Komisař hodnotí vzorek na soutěži.
Aktéři: Komisař
Vstupní podmínky: Komisař je přihlášen. Soutěž, komise i vzorek jsou otevřeny.
Hlavní scénář:  <ol style="list-style-type: none"> <li>1. Případ užití začíná, když komisař klikne na tlačítko „Hodnotit“.</li> <li>2. Systém zobrazí hodnotící formulář, ten obsahuje výpis viditelných atributů vzorku, komponenty potřebné pro typ hodnocení dané soutěže a také tlačítka „Potvrdit“, „Vyřadit“, „Porovnat“.</li> <li>3. Dokud nejsou zadány všechny hodnoty: <ol style="list-style-type: none"> <li>3.1 Systém zobrazí hlášku o nutnosti vyplnění všech hodnot.</li> </ol> </li> <li>4. Systém uloží hodnocení a zobrazí porovnání hodnocení.</li> </ol>
Alternativní scénář: <b>Hodnotit vzorek: částečné vyplnění</b>  <ol style="list-style-type: none"> <li>1. Alternativní scénář začíná krokem 3.1 hlavního scénáře.</li> <li>2. Komisař přeruší vyplňování hodnot a opustí formulář.</li> <li>3. Systém uloží částečně vyplněné hodnocení.</li> </ol>

Tabulka 2.1: Use case 06 Hodnotit vzorek - podrobná specifikace

### 2.3.4.3 Organizátor

Následující výčet případů užití obsahuje akce dostupné organizátorovi soutěže. Mají mu umožnit snadnou správu soutěže a komisi.

- UC 13 - Spustit soutěž
- UC 14 - Uzavřít soutěž
- UC 15 - Spustit komisi
- UC 16 - Uzavřít komisi
- UC 17 - Změnit předsedu komise
- UC 18 - Editovat komisaře

- UC 19 - Aktivovat komisaře v komisi
- UC 20 - Deaktivovat komisaře v komisi
- UC 21 - Přidat komisaře do komise
- UC 22 - Odebrat komisaře z komise
- UC 23 - Zobrazit přehled soutěže

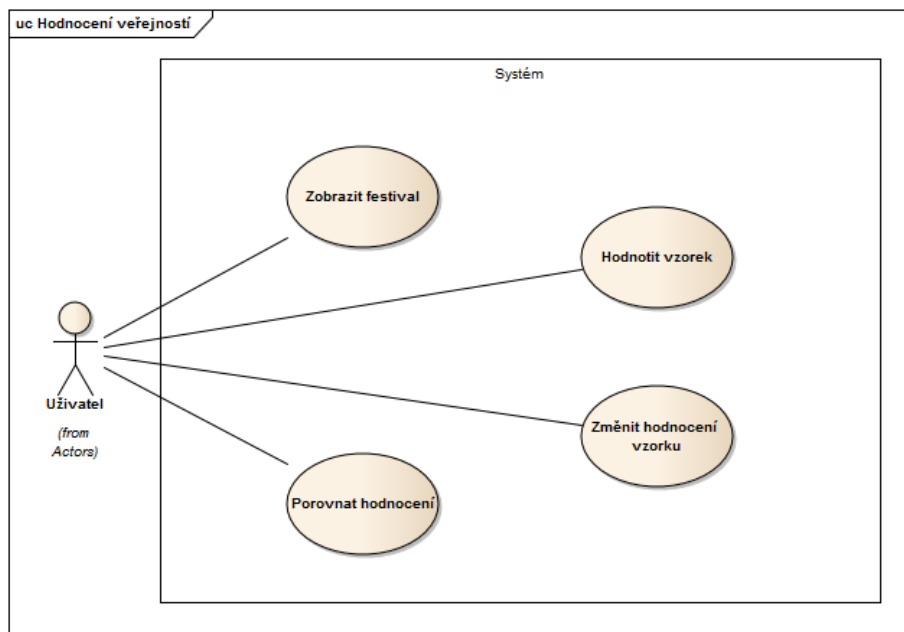
### 2.3.5 Hodnocení veřejností

Diagram případu užití je uveden na obrázku 2.4. V příloze B.3 jsou uvedeny podrobné specifikace případů užití.

#### 2.3.5.1 Uživatel

V následujících případech užití se pojmem uživatel myslí registrovaná a přihlášená osoba.

- UC 24 - Zobrazit festival
- UC 25 - Hodnotit vzorek
- UC 26 - Změnit hodnocení vzorku
- UC 27 - Porovnat hodnocení



Obrázek 2.4: Use case model - Hodnocení veřejností

## 2.4 Matice sledovatelnosti požadavků

Matice sledovatelnosti požadavků (anglicky requirements traceability matrix) je nástroj pro sledování požadavků vůči případům užití. Existuje-li požadavek, který není mapován na žádný případ užití, znamená to, že v modelu případu užití chybí. Platí to i obráceně. Existuje-li případ užití, který není mapován na žádný požadavek, znamená to, že množina požadavků není úplná [1]. Matice 2.2 potvrzuje, že všechny požadavky a případy užití jsou mapované.

		Use case																													
		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27			
Požadavky	01	x																													
	02		x																												
	03			x																											
	04				x																										
	05					x																									
	06						x																						x		
	07							x																					x		
	08								x																						
	09									x																					
	10										x																				
	11											x																			
	12												x																		
	13													x																	
	14														x																
	15															x															
	16																x														
	17																	x													
	18																		x												
	19																			x											
	20																					x									
	21																							x							
	22																								x						
	23																									x					
	24																										x				
	25																														
	26																														
	27																														
	28																														
	29																														
	30																														

Tabulka 2.2: Matice sledovatelnosti požadavků. V řádcích jsou uvedena čísla požadavků, ve sloupcích čísla případů užití.



## Kapitola 3

# Použité technologie

Následující kapitola krátce popisuje technologie, pomocí kterých byla práce vytvořena. Zahrnuje programovací jazyk, frameworky a doplňkové moduly. Hlavní použité technologie, python a django, vybral vedoucí práce.

### 3.1 Python

Python je interpretovaný, objektově orientovaný programovací jazyk. Je vyvíjen jako open source projekt a je tedy velmi široce používán. Mezi jeho výhody patří to, že je velmi stručný a díky tomu čitelný. Python je vyšší jazyk a má vestavěnou podporu pro listy, slovníky a další struktury. Python podporuje moduly a balíčky což umožňuje lepší modularitu programu a znovupoužití částí kódu [13]. Python má také velmi podrobnou a přehlednou dokumentaci<sup>1</sup>. Projekt používá python ve verzi 3.4, která vyšla v březnu 2014 a dosud je to nejaktuálnější stabilní verze jazyka. Jednou z největších výhod pythonu je široká nabídka externích balíčků a rozšíření. Tyto balíčky usnadňují programování, protože poskytují všestranná rozhraní od vývoje webových stránek až k přístupu k databázím a mnoho dalších. Mezi tyto balíčky patří například framework Django.

### 3.2 Django

Django je vysokoúrovňový webový framework v jazyce Python, který umožňuje rychlý vývoj a čistý vzhled [7]. Je založen na architektuře MVC (model, view, controller), to znamená, že odděluje datové modely, uživatelské rozhraní a řídicí logiku. Django umožňuje integraci databází do programů a podporuje různé databázové systémy jako PostgreSQL<sup>2</sup>, MySQL<sup>3</sup>, Oracle<sup>4</sup> a SQLite<sup>5</sup>. V projektu je použit PostgreSQL, důvodem je především open source licence. Vývoj pomocí frameworku Django je velmi rychlý a to díky využití integrovaných služeb. Django se tak dokáže postarat o autentizaci uživatelů, administrační rozhraní, tvorbu

---

<sup>1</sup><https://www.python.org/doc/>

<sup>2</sup><http://www.postgresql.org/>

<sup>3</sup><http://www.mysql.com/>

<sup>4</sup><http://www.oracle.com/>

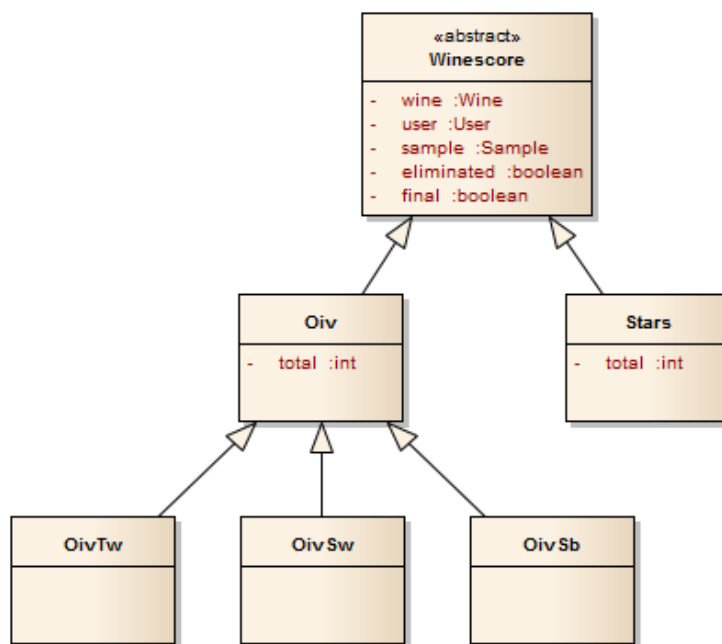
<sup>5</sup><http://www.sqlite.org/>



šablon, překlad stránek a další. A pokud Django samo o sobě nestačí, lze využít mnoho externích balíčků. Práce na projektu začala v Django verzi 1.6 a v průběhu se přešlo na verzi 1.7, především kvůli lepšímu API pro migraci datábázových schémat. Nejnovější verze 1.8 nabízí lepší kontrolu testovacích případů, ale kvůli nekompatibilitě některých externích balíčků se od použití této verze upustilo.

### 3.2.1 Polymorphic

Standardní práci s polymorfními modely v django zhoršuje fakt, že datábázové dotazy vrací kolekce nadřazených tříd. Django-polymorphic je doplňkový balíček frameworku django, který zjednodušuje dědičnost modelů. Dotaz na datábázi vrátí konkrétní instance tříd potomků [12]. V projektu je tato funkčnost využita pro generalizaci různých typů hodnocení. Třída rodiče „WineScore“ obsahuje definici atributů, které musí mít každé hodnocení. Tato třída je abstraktní, protože sama o sobě neobsahuje logiku a nechceme ji nikdy instanciovat. Od této třídy pak dědí konkrétní typy hodnocení (OIV, hvězdičky). Stupnice OIV má navíc další potomky, které se liší typem hodnoceného vína. Na obrázku 3.1 je uveden model tříd hodnocení.



Obrázek 3.1: Model tříd hodnocení.

Výhoda použití polymorfních modelů je demonstrována ve výpisu kódu 3.1. Dotaz do databáze vrátí konkrétní instance tříd potomků. Stejný dotaz, ale s využitím pouze nativního prostředí django, vrátí objekty nadřazené třídy (viz výpis kódu 3.2).

Balík django-polymorphic není jediný, který by řešil problém s polymorfními modely.

Listing 3.1: Výsledek dotazu s využitím polymorphic vrací konkrétní třídy

```
Oiv.objects.all()
[ <OivTw: id 1, total "85">,
  <OivSw: id 2, total "82">,
  <OivSb: id 3, total "91"> ]
```

Listing 3.2: Výsledek dotazu bez polymorphic vrací nadřazenou třídu

```
Oiv.objects.all()
[ <Oiv: id 1, total "85">,
  <Oiv: id 2, total "82">,
  <Oiv: id 3, total "91"> ]
```

Možností bylo více, například `django-polymorphic-models`<sup>6</sup> nebo `django-polymodels`<sup>7</sup>. Důvodem výběru prvního balíčku byla lepší dokumentace, elegantní a jednoduché řešení a také to, že je stále aktualizován.

### 3.2.2 Captcha

Captcha je zabezpečovací prvek k odlišení skutečných uživatelů od robotů. Zná ho asi každý, kdo si někdy registroval účet nebo psal na fórum. V projektu je tento prvek použit pro registraci nového komisaře do komise. Na výběr bylo mnoho balíčků, které implementují tuto funkčnost. Na stránkách `djangopackages`<sup>8</sup> je k dispozici přehledná tabulka s porovnáním balíčků. Pro projekt byl vybrán balíček `django-simple-captcha`<sup>9</sup>, který jako jeden z mála podporuje python verzi 3. Jeho použití je velmi jednoduché, ale zároveň nabízí širokou nastavitelnost. Například lze změnit velikost písma a nebo použít jednoduché matematické výrazy místo slov.

## Captcha



Obrázek 3.2: Ukázka vygenerované captcha. Uživatel musí opsat písmena z obrázku.

<sup>6</sup><https://code.google.com/p/django-polymorphic-models/>

<sup>7</sup><https://pypi.python.org/pypi/django-polymodels>

<sup>8</sup><https://www.djangopackages.com/grids/g/captcha/>

<sup>9</sup><http://django-simple-captcha.readthedocs.org/en/latest/>



# Kapitola 4

## Návrh

### 4.1 Návrh řešení

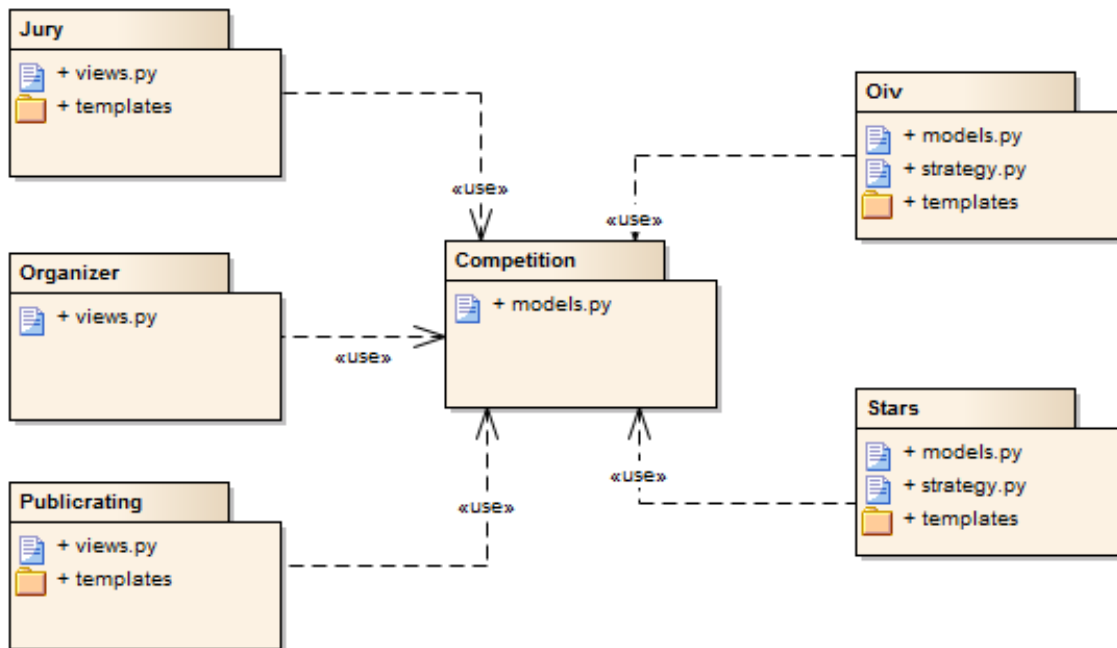
Důležitým prvním krokem návrhu řešení je navrhnout správnou strukturu projektu. Správná struktura je základem pro čistý, efektivní a znovupoužitelný kód. Python nabízí jednoduché řešení a to pomocí modulů a balíčků (anglicky packages). Pod pojmem modul se skrývá pouhé oddělení kódu do různých souborů a jeho účel je například oddělit části kódu zajišťující funkcionalitu a části udržující data. Následná spolupráce mezi těmito soubory je provedena pomocí příkazu `import`. Balíčky jsou v podstatě rozšíření mechanismu modulů do složek. Každá složka v projektu, která obsahuje soubor „`__init__.py`“ je považována za balíček. Model rozdělení aplikace do balíčků je uveden na obrázku 4.1. Balíčky použité v aplikaci se dají rozdělit do dvou kategorií: datové a funkční. Datové balíčky obsahují definice modelů a často také šablony vztahující se k těmto modelům. Naopak funkční balíčky obsahují definice metod, které pracují s těmito modely.

#### 4.1.1 Konfigurace soutěže

Konfigurace soutěže je řešena v balíčku „Organizer“. Tento balíček se, jak název napovídá, zabývá logikou aplikace z pohledu organizátora. Definice modelu soutěže je uvedena v balíčku „Competition“. Tato definice je rozšířena o nové atributy, potřebné pro konfiguraci soutěže. Je potřeba přidat pole pro logo soutěže, typ hodnocení a způsob udělování medailí. Hodnocení pomocí hvězdiček vyžaduje další atribut a to maximální počet udělovaných hvězdiček. Pro potřeby nastavení viditelných atributů vzorků je přidán nový model nazvaný „CompetitionSettings“. Ten obsahuje informace o tom, které atributy vína se budou zobrazovat během hodnocení vzorku.

#### 4.1.2 Průběh soutěže

Průběh soutěže je rozdělen do dvou balíčků. Hodnotící část pro komisaře a předsedy komise je v balíčku „Jury“ a organizační část je v balíčku „Organizer“. Dosavadní implementace hodnocení vzorku byla řešena jednou velmi dlouhou metodou. K rozštěpení této metody je možné použít Class-based views [3], což umožní rozdělení logiky pro http metody POST a GET do vlastních metod. Důležitou otázkou také je, jak zpracovávat různé typy hodnocení.



Obrázek 4.1: Model balíčků aplikace

Je možné vytvořit abstraktní model hodnocení „Winescore“, který bude obsahovat veškeré společné atributy hodnocení, například vzorek, víno a uživatel. Od tohoto modelu pak konkrétní typy hodnocení dědí a doplní vlastní logiku. V aplikaci jsou realizovány dva typy hodnocení a to v balíčku „OIV“ a „Stars“. Tyto balíčky obsahují definice modelu hodnocení, template html soubory a strategické metody. Tyto metody jsou používány aplikací bez ohledu na konkrétní typ hodnocení viz kapitola 4.3. Toto oddělení typů hodnocení do vlastních balíčků a použití strategie umožňuje velmi jednoduché přidání nových typů hodnocení. Není potřeba nijak měnit logiku průběhu soutěže.

### 4.1.3 Hodnocení veřejností

Hodnocení veřejností je další část, která má vlastní balíček a to „Festival“. Proces hodnocení vzorku na festivalu je velmi podobný tomu na soutěži, je tedy možné opět použít Class-based views s metodami post a get. Protože hodnocení veřejností by mělo být snadné a intuitivní, navrhuji použít menší počet obrazovek a sloučit funkčnost. Úvodní stránka festivalu zobrazuje tabulku se seznamem vzorků, ze které je možné dostat se na hodnocení vzorku, porovnání hodnocení a změnu hodnocení. Uživatel je zřejmé, které vzorky už hodnotil a jaké je jeho hodnocení a má možnost si vzorky seřadit.

## 4.2 Datový model

Pro zachycení struktury navrhovaného systému z předchozích odstavců je použit konceptuální model tříd. Tento model představuje statický pohled na modelovaný systém a jeho úkolem je znázornit typy objektů a jejich vztahy. Chtěl bych podotknout, že následující model na obrázku 4.2 není modelem celého systému, ale zaměřuje se pouze na části řešené touto prací. U jednotlivých tříd jsou uvedeny pouze klíčové atributy.

Třída "**Competition**" obsahuje atributy potřebné pro nastavení typu hodnocení (scoring type), způsobu udělování medailí (medal type), změnu loga. Dále obsahuje atribut pro určení škály hvězdiček (max\_stars) a zda se medaile udělují pouze jednomu vítězi (medal\_to\_one\_winner). Třída „CompetitionSettings“ pomáhá s nastavením soutěže a to konkrétně tak, že udržuje informace o tom, které atributy vín se mají zobrazovat při hodnocení.

Asociační třída "**JuryMember**" poskytuje vazbu M:N mezi uživatelem a komisí. Je to proto, že uživatel může být komisař ve více komisích. Také obsahuje atributy o tom, zda je uživatel předseda komise (is\_chair) a zda aktivně hodnotí (is\_active).

Třída "**Winescore**" je abstraktní a polymorfni. Je to proto, že různé typy hodnocení znamenají různé typy výsledků. Pomocí polymorfismu můžeme využít společné vlastnosti všech typů hodnocení a to například zda je hodnocení konečné (final) anebo zda je od prověřeného uživatele (verified). Tato třída je abstraktní, protože sama o sobě nemá hodnotu a tedy jí nechceme instanciovat.

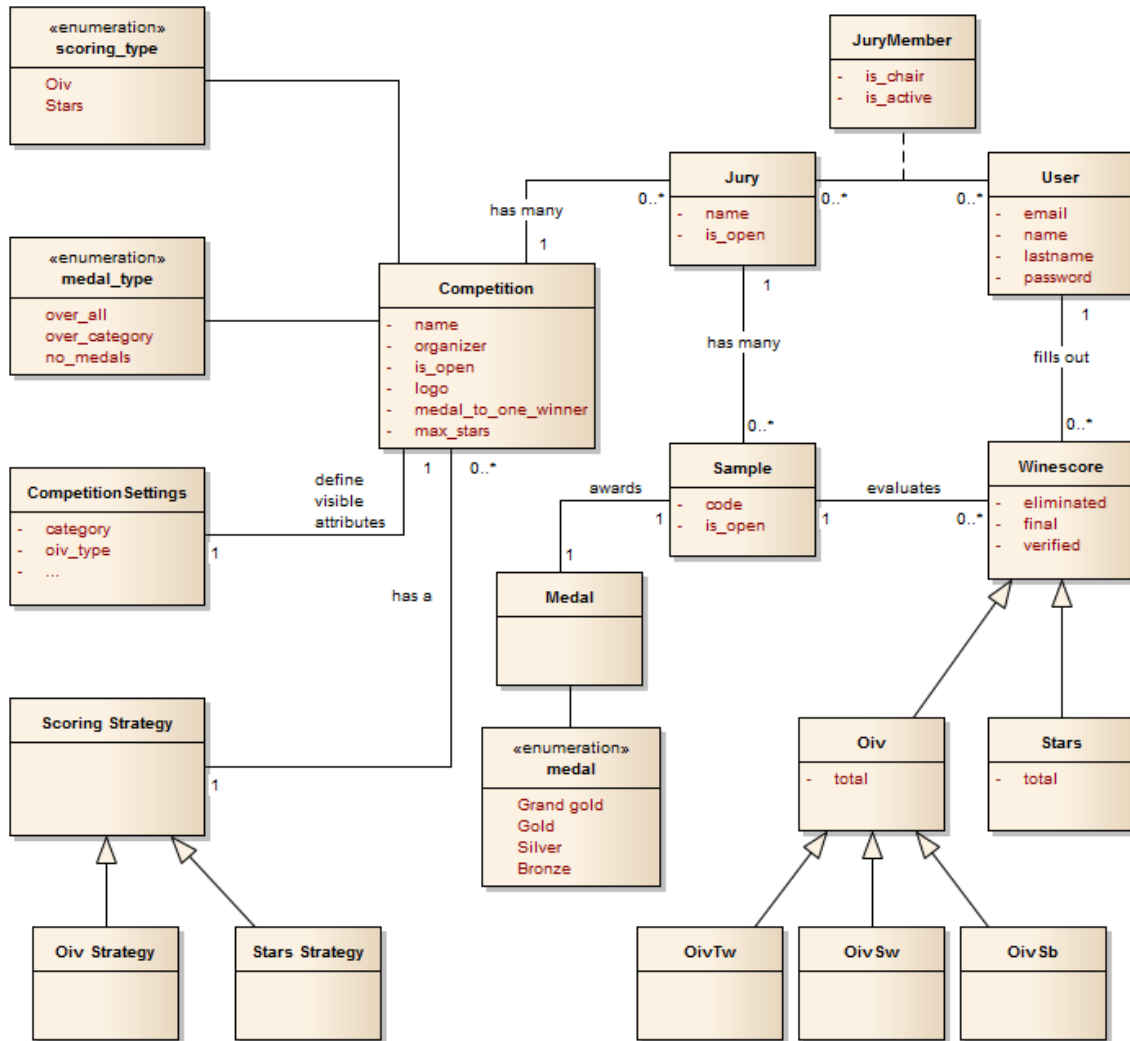
Třída "**Result**" má za úkol uchovávat průměrná hodnocení vzorků. Protože se v programu vyskytuje několik typů průměrů a také se tyto průměry zobrazují v několika případech, není dobré je pokaždé počítat znovu. Proto vznikla tato třída, které je opět rodičem pro konkrétní typy hodnocení. Průměry se počítají pouze po odeslání finálního hodnocení vzorku.

## 4.3 Použití návrhových vzorů

Návrhový vzor představuje obecné řešení problému, které se využívá při návrhu počítačových programů. Návrhový vzor není knihovnou nebo částí zdrojového kódu, která by se dala přímo vložit do našeho programu. Jedná se o popis řešení problému nebo šablonu, která může být použita v různých situacích. Objektově orientované návrhové vzory typicky ukazují vztahy a interakce mezi třídami a objekty, aniž by určovaly implementaci konkrétní třídy [11].

Možnost několika různých typů hodnocení vzorků<sup>1</sup> znamená vymyslet způsob, jak využít co nejvíce společného kódu. Logika hodnocení je prakticky stejná pro různé typy až na konkrétní zpracování hodnotícího formuláře. Protože rozhodování o tom, které hodnocení se použije, probíhá za běhu programu, nabízí se použití návrhového vzoru strategy [10]. Tento způsob umožňuje vybrat správnou logiku za běhu programu a tím pádem efektivně využít společný kód. V praxi to znamená, že každý typ hodnocení má svojí vlastní strategii, do které se přidávají části kódu specifické pro dané hodnocení. Další výhodou tohoto návrhového vzoru je to, že přidání dalších typů hodnocení je velice snadné. Stačí pouze doplnit logiku do strategických metod a případně vytvořit šablony pro grafické znázornění.

<sup>1</sup>OIV stupnice, hvězdičky



Obrázek 4.2: Konceptuální model tříd

# Kapitola 5

## Implementace

Kapitola Implementace popisuje výslednou podobu kódu programu. Struktura aplikace je uvedena v příloze C.1 a tato kapitola se řídí touto strukturou a popisuje význam vybraných souborů. Na tomto místě bych chtěl připomenout, že mým úkolem bylo poskytnout back end pro tuto aplikaci a proto design na použitých screenech aplikace není mou prací.

### 5.1 Modul Competition

Účelem balíčku "Competition" je definice modelů využívaných v rámci soutěže.

```
Competition
├── models.py
└── views.py
```

#### 5.1.1 Model balíčku Competition

Balíček "Competition" je typickým příkladem datového balíčku. Obsahuje klíčový soubor "models.py". Tento soubor obsahuje definice většiny modelů týkajících se soutěže. Na obrázku C.1 je zobrazen model tříd. Čárkovaný rámeček označuje třídy, jejichž definice se nachází v tomto balíčku. Důležitá je funkce "strategy", ta umožňuje generický průběh soutěže, bez ohledu na to jaký je její typ hodnocení. Tělo této funkce je uvedeno ve výpisu kódu 5.1.

Listing 5.1: Metoda strategy, vrací strategii podle způsobu hodnocení vzorků

```
@property
def strategy(self):
    if self._strategy:
        return self._strategy
    if self.scoring_type == "Oiv":
        from Oiv.strategy import OivStrategy
        self._strategy = OivStrategy()
        return self._strategy
    if self.scoring_type == "Stars":
        from Stars.strategy import StarsStrategy
        self._strategy = StarsStrategy()
        return self._strategy
```



Definice konkrétních strategií se nacházejí ve svém vlastním balíčku (Oiv, Stars). Tato metoda má před názvem anotaci `@property`. Je to funkčnost pythonu, jak zajistit privátnost proměnné (ekvivalent funkce `get` v Javě), ale také umožňuje vytvářet vlastní logiku proměnných. Takto anotované metody se totiž chovají jako proměnné co se týče jejich volání [14]. Soutěž má tedy proměnnou "`strategy`", díky které přistupuje ke specifickým funkcím jednotlivých způsobů hodnocení. Příkladem těchto funkcí je získání šablony nebo průměrných hodnot hodnocení.

### 5.1.2 Implementovaná funkčnost balíčku `Competition`

Balíček "`Competition`" neobsahuje žádné vlastní obrazovky. V souboru "`views.py`" je pouze metoda "`index`", která přesměrovává uživatele na patřičnou stránku podle jeho uživatelské role, ale také podle dalších kritérií. Například je potřeba rozlišit, zda jsou v soutěži otevřené komise a vzorky.

## 5.2 Modul `Jury`

Balíček "`Jury`" obaluje funkčnost komisí a komisařů na soutěžích. Neobsahuje žádné definice modelů, ty jsou uvedeny v balíčku "`Competition`". Součástí tohoto balíčku je složka "`templates`", která zahrnuje html soubory. Framework `django` obsahuje templatovací jazyk, který umožňuje přidávat prezentační logiku do html souborů [8]. Lze tak například měnit vzhled podle uživatelských rolí, předseda komise má k dispozici více tlačítek, ale není třeba vytvářet jedinečný template jen pro předsedu komise.

```
Jury
├── templates
├── urls.py
└── views.py
```

### 5.2.1 Implementovaná funkčnost balíčku `Jury`

V souboru "`views.py`" jsou definice metod kontrolující průběh soutěže. Dále uvádím jejich výčet a které případy užití implementují.

#### **EditView - post, get:**

##### **Realizované use case:**

*UC 06 - Hodnotit vzorek*

*UC 07 - Vyřadit vzorek*

**Popis:** Hodnotící formulář obsluhuje třída "`EditView`", která obsahuje metody `post` a `get`. Jedná se o `Class-based view` [3]. Obě metody využívají strategické metody pro obsluhu prvků, které jsou specifické pro různé typy hodnocení. Příklad získání formuláře je uveden ve výpisu kódu 5.2.

Listing 5.2: Získání formuláře pomocí `strategy`

```
form = sample.jury.competition.strategy.get_form(wine_score)
```

Hodnotící formuláře jsou definované v balíčcích podle typu hodnocení, které realizují. Využívají takzvané modelové formuláře, což je funkčnost djanga umožňující vytvářet formuláře pro konkrétní modely [6]. Práce s takovými formuláři je velmi snadná, není třeba překládat jednotlivé formulářové pole na atributy modelů. Ukázka práce s tímto formulářem, který uloží vyplněné skóre, je uvedena ve výpisu kódu 5.3.

Listing 5.3: Zpracování formuláře

```
form = StarsForm(request.POST, instance=wine_score)
if form.is_valid():
    form.save()
```

**chair\_index:**

**Realizované use case:**

*UC 09 - Zavřít vzorek*

*UC 10 - Otevřít vzorek*

*UC 11 - Zobrazit přehled komise*

**Popis:** Metoda obsahuje logiku přesměrování, podle toho, zda je uživatel opravdu předseda komise a zda je komise otevřená. Pro snazší identifikaci role uživatele se používají metody definované v souboru "acl.py". Další logika je ukryta uvnitř html souboru pomocí templatovacího jazyka. Například ovládání otevírání a zavírání vzorku je realizováno v souboru "chair\_menu.html" a to tak, jak je uvedeno ve výpisu kódu 5.4.

Listing 5.4: Pseudokód otevření a zavření vzorku

```
if existuje otevreny vzorek:
    if vzorek není ohodnocen predsedou:
        zobraz tlacitko Hodnotit
    if vzorek není ohodnocen vsemi komisari:
        tlacitko Potvrdit je deaktivovane
    else:
        tlacitko Potvrdit je aktivovane a uzavre vzorek
else:
    if existuje neohodnoceny vzorek:
        zobraz tlacitko Otevrit vzorek
```

Z této logiky pak vyplývají tři možnosti, které jsou vyobrazeny na obrázku 5.1. Vlevo je tlačítko "Potvrďte" deaktivované, protože chybí hodnocení jiného komisaře. Uprostřed je také deaktivované, ale přibylo tlačítko "Hodnotit", protože chybí hodnocení předsedy komise. Vpravo pak jsou splněny podmínky a je možné vzorek uzavřít.

**winescore\_correct:**

**Realizované use case:**

*UC 12 - Povolit změnu hodnocení*

**Popis:** Metodu spustí předseda komise z přehledu komise a to kliknutím na tlačítko "Upravit" u jména komisaře. Metoda tomuto komisaři umožní změnit své hodnocení a to tak, že nastaví atribut final=False u daného objektu "Winescore".



Obrázek 5.1: Tlačítka pro zavření vzorku

`sample_close`:

**Realizované use case:**

*UC 09 - Zavřít vzorek*

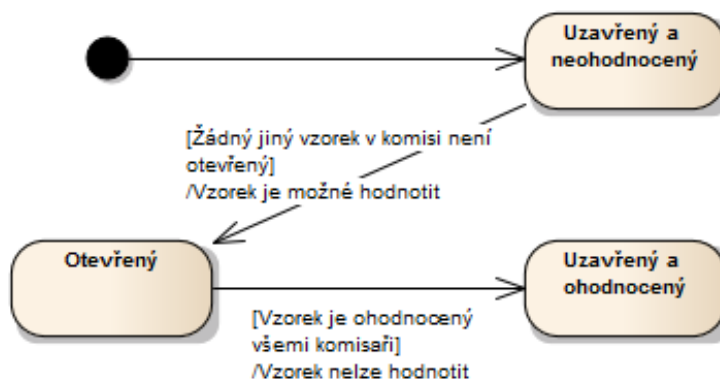
**Popis:** Metoda kontroluje, zda jsou splněny všechny podmínky na zavření vzorku. Vzorek lze zavřít, pokud byl ohodnocen všemi aktivními komisaři v komisi.

`sample_open`:

**Realizované use case:**

*UC 10 - Otevřít vzorek*

**Popis:** Tato metoda slouží pro otevření dalšího vzorku v komisi. Stavový diagram vzorku (Sample) je uveden na obrázku 5.2.



Obrázek 5.2: Stavový diagram vzorku

`list_samples`:

**Realizované use case:**

*UC 08 - Porovnat své hodnocení*

**Popis:** Komisař má možnost během hodnocení porovnat svá předchozí hodnocení. Tato hodnocení jsou seřazena sestupně podle času uzavření vzorku, tedy nejdříve se zobrazují nejaktuálnější hodnocení.

## 5.3 Modul Oiv

Tento balíček obaluje modely a logiku typu hodnocení OIV. OIV definuje tři typy hodnocení, které se liší v několika atributech vína, podle kterých se hodnotí. Tyto typy mají pevně stanovenou škálu hodnocení. Na soutěžích typu OIV se také udělují medaile pro vzorky s výjimečnými výsledky.

```
Oiv
├── medals.py
├── models.py
├── model_forms.py
└── strategy.py
```

### 5.3.1 Udělování medailí

V souboru "medals.py" se nachází třída "OivMedalGrant", která obsahuje metody pro udělování medailí. Medaile se udělují pouze na soutěžích s typem hodnocení Oiv. Existují čtyři druhy medailí: Grand Gold, Gold, Silver, Bronze a jejich udělování se řídí metodikou OIV [9]. Medaile se udělují prvním 30 procentům vzorků, ale je také možné zvolit pouze absolutního vítěze soutěže. O tom rozhoduje nastavení soutěže, konkrétně atribut medal\_to\_one\_winner.

### 5.3.2 Modely balíčku Oiv

Třída "Oiv" je potomek třídy "Winescore", dědí od ní atributy jako wine, user, sample a další. Sama přidává atribut total a metody pro jeho počítání. Konkrétní atributy, podle kterých se vína hodnotí, jsou definovány ve třídách "OivTw", "OivSw" a "OivSb". Tyto třídy dědí od třídy "Oiv", jsou to tedy modely pro uložení hodnocení vzorků. Pro lepší pochopení významu těchto tříd uvedu ukázkou hodnocení. Komisař hodnotí vzorek podle atributů jako čirost, čistota, kvalita a pro každý tento atribut má pevně danou bodovou škálu. Počet bodů těchto jednotlivých atributů se uloží do objektu "OivTw" a součet těchto bodů se uloží do atributu total objektu "Oiv".

Poslední třídou definovanou v tomto souboru je "OivResult". Je to potomek třídy "Result" a obsahuje atribut avg, oiv\_avg a trim\_avg, což jsou průměrné hodnocení vzorku. Tento objekt má tedy vazbu na třídu "Sample" a definuje několik typů průměrných hodnocení daného vzorku. Tyto průměry se počítají po každém uložení hodnocení vzorku. Avg označuje aritmetický průměr hodnocení. Trim\_avg označuje aritmetický průměr bez nejmenší a největší hodnoty. Je možné ho počítat, pouze pokud počet hodnocení je větší nebo rovno třem, jinak je stejný jako avg. Oiv\_avg se počítá tak, že se nejprve spočítá aritmetický průměr, pak se vyřadí všechny vzorky, které se od tohoto průměru odchyľují o 7 bodů a více a ze zbylých vzorků se opět spočítá aritmetický průměr.

### 5.3.3 Formuláře

Soubor "model\_forms.py" obsahuje definice tří modelových formulářů pro hodnocení [6]. Vycházejí ze tříd "OivTw", "OivSw" a "OivSb" a obsahují pole pro atributy vzorku, které se

uloží při uložení formuláře. Na obrázku 5.3 je hodnoticí tabulka pro nešumivá vína (OivTw), jsou vidět bodové škály pro jednotlivé atributy a také vysvětlení významu těchto atributů.

Komise: Jury 1, Vzorek: 1  
Ročník sklizně: 1990

Čírost	5	4	3	2	1
Vzhled mimo čírosti	10	8	6	4	2
Čistota	6	5	4	3	2
Pozitivní intenzita	8	7	6	4	2
Kvalita	16	14	12	10	8
Čistota	6	5	4	3	2
Pozitivní intenzita	8	7	6	4	2
Harmonická perzistence	8	7	6	5	4
Kvalita	22	19	16	13	10
Harmonie – Celkový dojem		10	9	8	7
Celkem			43		

Potvrďte Vyřadit Já Všichni

Odpovídající celkovému posouzení produktu

Obrázek 5.3: Hodnoticí formulář OIV

### 5.3.4 Strategie

Tento soubor obsahuje třídu "OivStrategy", což je jedna ze strategií využívaná soutěží v rámci návrhového vzoru strategy, který je popsán v sekci o použití návrhových vzorů 4.3. Tato třída implementuje funkce, které obsluhují objekty a soubory specifické pro hodnocení OIV stupnicí. Implementace soutěže pak nemusí rozlišovat, mezi jednotlivými typy hodnocení, to za ní obstará strategie. Příklad strategické metody je uveden ve výpisu kódu 5.5.

Listing 5.5: Strategická metoda pro získání template

```
def get_template_show_results(self):
    return 'oiv_show_results.html'
```

Tato funkce vrací název templatovacího html souboru pro hodnocení OIV. Jiné typy hodnocení také implementují tuto metodu, ale vracejí soubor pro dané hodnocení.

## 5.4 Modul Organizer

Balíček "Organizer" zahrnuje velké množství funkcí týkajících se organizace a průběhu soutěží z pohledu organizátora. Implementuje veškeré případy užití uvedené v kapitole Analýza pro aktéra Organizátor viz výčet use case v sekci 2.3.3.1 a 2.3.4.3. Rozebírám zde dva soubory, "views.py" obsahuje metody a "forms.py" obsahuje definice použitých formulářů.

```
Organizer
├─ views.py
```

forms.py

### 5.4.1 Implementovaná funkčnost balíčku Organizer

`competition_settings`:

**Realizované use case:**

*UC 01 - Změnit typ hodnocení soutěže*

*UC 02 - Změnit logo soutěže*

*UC 03 - Vybrat viditelné atributy vína*

*UC 04 - Nastavit způsob udělování medailí*

*UC 05 - Nastavit počet hvězdiček pro hodnocení*

**Popis:** Tato metoda řeší nastavení atributů soutěže, využívá formulář `CompetitionSettingsForm` a ten je pro přehlednost rozdělen do tří záložek. Logo se mění vybráním požadovaného souboru pomocí pole `FileField` [5]. Změnit typ hodnocení soutěže je možné, jen pokud neexistuje žádné hodnocení a výběrové tlačítko je deaktivované pomocí javascriptu, pokud typ hodnocení měnit nelze. Počet udělovaných hvězd je také možné měnit pouze tehdy, když neexistuje hodnocení, ale navíc je pole aktivní jen při typu hodnocení Stars. Na obrázku 5.4 je screen z editačního formuláře. K výběru viditelných atributů vín je přidána tabulka pěti vzorků s výpisem jejich atributů, aby bylo zřejmé, co jednotlivé atributy představují.

Obrázek 5.4: Formulář nastavení soutěže, záložka nastavení.

`competition_open`:

**Realizované use case:**

*UC 13 - Spustit soutěž*

**Popis:** Metoda spustí soutěž. Nastaví atribut `is_open=True`.

#### `competition_close:`

**Realizované use case:**

*UC 14 - Uzavřít soutěž*

**Popis:** Metoda uzavře soutěž a veškeré její komise. Pokud je soutěž typu OIV tak také udělí medaile (viz 5.3.1).

#### `competition_jury_open:`

**Realizované use case:**

*UC 15 - Spustit komisi*

**Popis:** Metoda spustí komisi. Nastaví atribut `is_open=True`. V otevřené komisi mohou komisaři hodnotit.

#### `competition_jury_close:`

**Realizované use case:**

*UC 16 - Uzavřít komisi*

**Popis:** Metoda uzavře komisi. V uzavřené komisi nelze hodnotit.

#### `edit_jury:`

**Realizované use case:**

*UC 17 - Změnit předsedu komise*

**Popis:** Metoda změní předsedu komise. Předseda se mění kliknutím na tlačítko u daného komisaře, na obrázku 5.5 je aktuální předseda zobrazen tučně a jeho tlačítko je vybrané.

Uživatelské jméno	Předseda poroty	Upravit
k1@vinarroku2014.cz	<input checked="" type="radio"/>	
k2@vinarroku2014.cz	<input type="radio"/>	

Obrázek 5.5: Formulář editace komise.

#### `edit_jury_edit_member:`

**Realizované use case:**

*UC 18 - Editovat komisaře*

**Popis:** Organizátor může změnit jméno, příjmení a heslo komisaře. Do editačního formuláře se dostane kliknutím na ikonu za emailem komisaře, na obrázku 5.5 ikona tužky.

**edit\_jury\_activate\_member:****Realizované use case:**

*UC 19 - Aktivovat komisaře*

**Popis:** Pokud byl komisař deaktivován, nemůže hodnotit. Organizátor může komisaře aktivovat z přehledu komise.

**edit\_jury\_delete\_member:****Realizované use case:**

*UC 20 - Deaktivovat komisaře*

*UC 22 - Odebrat komisaře z komise*

**Popis:** Na obrázku 5.5 lze kliknutím na křížek ve sloupci Upravit a v řádku daného komisaře smazat či deaktivovat komisaře. Komisaře lze smazat, pouze pokud zatím neohodnotil žádný vzorek v komisi. Pokud takové hodnocení existuje, komisaře lze pouze deaktivovat. Deaktivovaný komisař nemůže hodnotit, ale komise nemusí čekat na jeho hodnocení.

**edit\_jury\_add\_member:****Realizované use case:**

*UC 21 - Přidat komisaře do komise*

**Popis:** Tato metoda realizuje přidání nového komisaře do komise. Na obrázku 5.5 se kliknutím na tlačítko "Přidat nového komisaře" dostane organizátor do formuláře, který je stejný jako registrace uživatele. Nově vytvořený uživatel je pak přiřazen jako komisař do komise.

**edit\_jury\_add\_member\_select:****Realizované use case:**

*UC 21 - Přidat komisaře do komise*

**Popis:** Tato metoda realizuje přidání komisaře z jiné komise v rámci jedné soutěže. Lze přidat komisaře pouze z jiných komisí a to kliknutím na tlačítko „Přidat komisaře z jiné komise“ na obrázku 5.5.

**samples\_show:****Realizované use case:**

*UC 23 - Zobrazit přehled soutěže*

**Popis:** Přehled soutěže je ukázán na obrázku 5.6, jedná se o soutěž hodnocenou hvězdičkami a proto se zde vyskytuje pouze aritmetický průměr. Soutěže hodnocené OIV stupnicí mají navíc ořezaný průměr a OIV průměr. Tabulka je plněná pomocí technologie AJAX, není tedy potřeba obnovovat stránku. Je možné vyhledávat podle názvu vína a řadit řádky.



Vzorek	Komise	Popis vína	Ocenění	Medaile	Stav hodnocení	Aritmetický průměr
1	Jury 1	ZVino 1, Vinařství 1, 1990, Suché, 10	1 ★	0 ★	2/2	1.00
2	Jury 1	Vino 2, Vinařství 2, 1991, Suché, 11	0 ★	0 ★	2/2	3.00
3	Jury 1	Vino 3, Vinařství 3, 1992, Suché, 12	0 ★	0 ★	3/2	4.50

Showing 1 to 3 of 3 entries  
 Show 10 entries  
 Předchozí 1 Další

Obrázek 5.6: Přehled soutěže.

### 5.4.2 Formuláře

Soubor "forms.py" obsahuje definice dvou formulářů užitých ve výše popsáných metodách. CompetitionSettingsForm obsahuje všechny pole potřebné pro nastavení soutěže. Jsou to pole pro logo, typ hodnocení soutěže, typ udělování medailí, jeden vítěz, maximální počet hvězd a nakonec pole pro všechny atributy vína. EditMemberForm je použit pro editaci komisařových údajů. Heslo je potřeba zadat dvakrát a validace, zda se hesla shodují, je implementovaná pomocí metody "clean\_password\_confirm". Django umožňuje validovat jednotlivé prvky formuláře a tato metoda bude volána automaticky při kontrole validity formuláře [4].

## 5.5 Modul Festival

Tento balíček se zabývá hodnocením veřejností na takzvaných festivalech. Festival, stejně jako soutěž, má určený typ hodnocení (OIV, hvězdy), ale vzorky hodnotí široká veřejnost.

```
Festival
├─ views.py
```

### 5.5.1 Implementovaná funkčnost balíčku Festival

show\_all\_samples:

**Realizované use case:**

- UC 24 - Zobrazit festival
- UC 26 - Změnit hodnocení vzorku
- UC 27 - Porovnat hodnocení

**Popis:** Tato metoda přesměrovává uživatele na úvodní stránku festivalu viz obrázek 5.7. Ta obsahuje tabulku se seznamem vzorků, uživatelským hodnocením a s odkazy na hodnocení vzorků. Vlastní plnění, vyhledávání a řazení této tabulky zajišťuje metoda "get\_data". Změnit své předchozí hodnocení může uživatel provést kliknutím na ikonu tužky vedle svého dosavadního hodnocení. Porovnat hodnocení s hodnoceními oficiálních komisařů je možné kliknutím na popis vína daného vzorku. Zobrazí se stránka s krátkým přehledem vzorku a s tabulkou hodnocení.



Obrázek 5.7: Úvodní strana festivalu.

**RatingView:****Realizované use case:***UC 25 - Hodnotit vzorek*

**Popis:** Hodnocení vzorků na festivalu je podobné tomu na soutěžích. Opět je využito class based view [3] s metodami post a get. Také se zde využívají strategické metody pro různé typy hodnocení. Rozdíl oproti hodnocení na soutěže je například to, že vzorky nelze vyřadit a také není potřeba rozlišovat komise a předsedy komisí.

## 5.6 Modul Stars

Balíček "Stars" je velmi podobný balíčku "Oiv". Oba dva popisují typ hodnocení vzorků. Hodnocení pomocí hvězdiček je realizováno především pro užití na festivalech, protože je jednoduché a rychlé.

```
Stars
├── models.py
├── model_forms.py
└── strategy.py
```

### 5.6.1 Modely balíčku Stars

Třída "Stars" je potomkem třídy "Winescore". Je to tedy hodnocení s atributy jako wine, sample, eliminated a přidává atribut total, což je počet udělených hvězd. Dále je v tomto souboru definovaná třída "StarsResult", potomek třídy "Result" a obsahuje průměr stars\_avg, což je průměrné hodnocení vzorku. Pro hodnocení hvězdami není potřeba jiných průměrů, ale jejich přidání je možné.

### 5.6.2 Formuláře

Tento soubor obsahuje definici hodnotícího formuláře "StarsForm". Je to modelový formulář [6] podle modelu "Stars" a je napsaný tak, aby bylo možné ho použít na libovolný počet hvězd. V nastavení soutěže je možné zvolit maximální počet hvězd a ten je omezen na rozmezí 5 až 15.

### 5.6.3 Strategie

V tomto souboru je definována třída "StarsStrategy", což je strategie pro typ hodnocení. Její OIV ekvivalent je třída "OivStrategy" popsaná zde 5.3.4. Strategické metody obsahují implementaci specifickou pro hodnocení hvězdami.

Listing 5.6: Strategická metoda pro získání template

```
def get_template_show_results(self):  
    return 'stars_show_results.html'
```

## 5.7 Zhodnocení implementace

Povedlo se implementovat všechny případy užití deklarované v kapitole analýza 2.3.2. Formulář pro konfiguraci soutěže, byl v průběhu implementace rozdělen do záložek (viz obrázek 5.4). To hodnotím pozitivně, formulář je mnohem přehlednější a je jasné, kde se jaké položky nacházejí. Na druhou stranu formulář obsahuje tabulku s přehledem atributů vín pěti vzorků. Účelem této tabulky je pomoci organizátorovi s výběrem viditelných atributů vín pro hodnocení. Bohužel je tato tabulka velmi široká a nepraktická. Možné řešení tohoto problému by bylo použít popover [2] pro jednotlivé položky, podobně jak je tomu u hodnocení vzorku.

Dále bych chtěl poukázat na fragmentaci kódu pro hodnocení vzorku. Původně tuto funkčnost zajišťovala jedna metoda, ale díky použití class based view [3] bylo možné metodu rozštěpit na část pro získání (GET) a odeslání (POST) stránky. To přineslo lepší čitelnost a efektivitu kódu.

Co se týče hodnocení veřejností, byla implementována veškerá funkcionalita. K čemu ale nedošlo, bylo vymezení festivalu do samostatného modelu. V aplikaci tedy festival vychází ze soutěže.

Nakonec bych chtěl ocenit přínos použití návrhového vzoru strategy. To umožnilo zobecnit logiku soutěž pro jakýkoli typ hodnocení. Systém se rozhoduje za běhu, které formuláře a které šablony má použít. Budoucí přidání nového typu hodnocení znamená vytvoření vlastního balíčku s šablonami, potřebnými modely a strategií. Není třeba zasahovat do logiky soutěže.

# Kapitola 6

## Testování

Testování je důležitou součástí vývoje software. Hlavními důvody testování je odhalit defekty, zamezit jejich opětovnému implementování a ověřit, zda vše funguje jak má. Mělo by probíhat po celou dobu vývoje. Automatizované testování se provádí pomocí testů, které lze pouštět opakovaně. Hlavní výhodou automatizace testů je časová úspora, snazší testování a lepší pokrytí. Jedním z využití těchto testů je ověření, že nové přidaná funkcionality či refactoring nenarušil dosavadní funkčnost.

### 6.1 Selenium

Selenium je široce rozšířený testovací nástroj používaný pro automatické testování webových aplikací [16]. Skládá se z několika nezávislých komponent. Selenium IDE je plugin do prohlížeče Mozilla Firefox a nabízí velmi jednoduchý způsob vytváření testů. Testy se nahrávají jako makra, stačí manuálně projít, to co chceme testovat. Takto vytvořené testy lze následně konfigurovat v přehledném IDE a také je možné tyto testy exportovat do několika programovacích jazyků. Mezi těmito jazyky je i python, ale pouze ve verzi 2 a tyto exportované testy nebylo možné použít. Další komponentou je Selenium WebDriver, což je nástroj, který umožňuje psaní těchto testů pomocí programovacích jazyků. Je nutné specifikovat v jakém prohlížeči chceme testy spouštět, v projektu je použit prohlížeč Firefox. Ve výpisu kódu 6.1 uvádím jednoduchý testovací případ, který testuje přihlášení uživatele.

Listing 6.1: Ukázka testovacího případu přihlášení uživatele

```
def test_user_can_login(self):
    self.browser.get(self.live_server_url + '/login/')
    self.browser.find_element_by_name("email").clear()
    self.browser.find_element_by_name("email").send_keys("vinar@email.cz")
    self.browser.find_element_by_name("password").clear()
    self.browser.find_element_by_name("password").send_keys("heslo")
    self.browser.find_element_by_id("submit").click()
    body = self.browser.find_element_by_tag_name('body')
    self.assertIn(u"vinar@email.cz", body.text)
```

Důvodem výběru tohoto typu testování byl fakt, že se jedná o webovou aplikaci. Je potřeba se ujistit, že vše funguje, tak jak má a to zejména z pohledu uživatele. Selenium

umožňuje, jak testování funkčnosti a průchodu aplikací, tak testování vzhledu. Lze otestovat to, že kliknutí na tlačítko vyvolá správnou akci a zároveň se ujistit, že tlačítko je vidět a má správnou barvu. Existují situace, kde Selenium nestačí, nebo je jeho použití zbytečné. Proto je vhodné doplnit testování o další typ testů, například o unit testy, které jsou užitečné pro testování dílčích částí kódu.

Testování pomocí frameworku selenium nabízí možnost takzvaného nahrávání a přehrávání, kdy se napodobuje manuální průběh testu, který se zachytí a může využít znovu. Bohužel takto vytvořené testy nebylo možné exportovat do prostředí aplikace a testy tedy byly psány ručně. Pokrytí kódu testy vyplývá z případů užití definovaných v kapitole analýza 2.3.2, ale bylo vytvořeno také mnoho pomocných testů. Většina testů vznikla v průběhu implementačních prací a tyto testy byly spouštěny po větších úsecích práce. Mezi nalezené defekty patří změny url po přesunu kódu mezi balíčky, chyby v uživatelském rozhraní a špatně fungující překlady. V testech se totiž používá jazyk angličtina, kdežto většina hlášek v programu je psána v češtině. Díky tomu testy poskytují kontrolu, že překlady existují. Finální testování proběhlo bez chyb a přehled je uveden v příloze D.

### 6.1.1 Jury

Struktura testů vychází z rozdělení aplikace do balíčků. V balíčku „Jury“ jsou testy průběhu soutěže z pohledu komisaře a předsedy komise. Jelikož se průběh soutěže, zvláště hodnocení vzorků, liší podle typu hodnocení, vznikly testy pro soutěž typu OIV i typu Stars. Přehled případů užití a testů, které je pokrývají je v tabulce 6.1 a screen těchto testů v příloze D.1.

Případ užití	Testovací případy
<b>Komisař</b>	
UC 06 - Hodnotit vzorek	test_jury_member_can_evaluate
UC 07 - Vyřadit vzorek	test_jury_member_can_eliminate_sample
UC 08 - Porovnat své hodnocení	test_jury_member_can_compare_his_evaluations
<b>Předseda komise</b>	
UC 09 - Zavřít vzorek	test_chair_can_close_sample
UC 10 - Otevřít vzorek	test_chair_can_open_sample
UC 11 - Zobrazit přehled komise	test_chair_can_see_scores_of_whole_jury
UC 12 - Povolit změnu hodnocení	test_chair_can_allow_edit_of_final_sample

Tabulka 6.1: Tabulka případů užití a testovacích případů balíčku „Jury“.

### 6.1.2 Organizer

V tomto balíčku jsou testy z pohledu organizátora a to jak testy konfigurace soutěže, tak testy průběhu soutěže. Typ soutěže, z hlediska organizace, nehraje roli a proto není třeba rozlišovat více testů. V tabulce 6.2 jsou uvedeny případy užití a testovací případy. Screen ze závěrečného testování je uveden v příloze D.2.

Případ užití	Testovací případy
<b>Konfigurace soutěže</b>	
UC 01 - Změnit typ hodnocení soutěže	test_scoring_type_change
UC 02 - Změnit logo soutěže	test_logo_change
UC 03 - Vybrat viditelné atributy vína	test_visible_attributes_change
UC 04 - Nastavit způsob udělování medailí	test_medal_type_change
UC 05 - Nastavit počet hvězdiček pro hodnocení	test_max_stars_change
<b>Průběh soutěže</b>	
UC 13 - Spustit soutěž	test_organizer_can_open_competition
UC 14 - Uzavřít soutěž	test_organizer_can_close_competition
UC 15 - Spustit komisi	test_organizer_can_open_jury
UC 16 - Uzavřít komisi	test_organizer_can_close_jury
UC 17 - Změnit předsedu komise	test_organizer_can_change_jury_chair
UC 18 - Editovat komisaře	test_organizer_can_edit_jury_member
UC 19 - Aktivovat komisaře v komisi	test_organizer_can_activate_jury_member
UC 20 - Deaktivovat komisaře v komisi	test_organizer_can_deactivate_jury_member
UC 21 - Přidat komisaře do komise	test_organizer_can_add_jury_member_from_other_jury, test_organizer_can_add_jury_new_member
UC 22 - Odebrat komisaře z komise	test_organizer_can_delete_jury_member
UC 23 - Zobrazit přehled soutěže	test_organizer_can_see_competition_overview

Tabulka 6.2: Tabulka případů užití a testovacích případů balíčku „Organizer“.

### 6.1.3 Festival

Pro veřejné hodnocení je hlavní typ hodnocení pomocí hvězdiček a proto je toto hodnocení použito v testech. Tabulka 6.3 obsahuje případy užití a testovací případy a zde D.3 je screen ze závěrečného testování.

## 6.2 Unit

Unit testy slouží pro testování samostatných částí kódu. Unit testy frameworku django vycházejí z knihovny python unittest [15]. V aplikaci jsou unit testy použity tam, kde by testování pomocí selenium testů bylo zbytečně složité. To nastává pro otestování správného počítání průměrných hodnot hodnocení. V aplikaci jsou tři typy průměrů a pro jejich otesto-

Případ užití	Testovací případy
UC 24 - Zobrazit festival	test_main_festival_page_shows_samples
UC 25 - Hodnotit vzorek	test_user_can_evaluate_festival_sample
UC 26 - Změnit hodnocení vzorku	test_user_can_reevaluate_festival_sample
UC 27 - Porovnat hodnocení	test_user_can_compare_evaluations

Tabulka 6.3: Tabulka případů užití a testovacích případů balíčku „Festival“.

vání pomocí selenium testu by bylo nutné vytvořit několik hodnocení. Unit test této jednotky je v souboru „tests\_unit.py“ v balíčku „Jury“.

## Kapitola 7

# Zhodnocení a budoucnost práce

### 7.1 Shrnutí práce

Práce na tomto projektu probíhaly podle plánu. Na počátku jsem se seznámil s dosavadní implementací a prostředím a mým úkolem bylo zlepšení organizace a editace soutěží. Současně s tím probíhala analýza navrhované funkcionality. Později jsem již implementoval přidání nového typu hodnocení pomocí hvězdiček, což vyžadovalo rozsáhlejší refactoring kódu, za účelem zobecnění průběhu soutěže. Realizace hodnocení veřejností se drobně změnila a finální verze úzce souvisí s klasickou soutěží.

V průběhu prací došlo na upgrade programovacího prostředí, konkrétně frameworku django na verzi 1.7. To přineslo zlepšení přidávání nových modelů a změny stávajících modelů pomocí migrací. Na druhou stranu nebylo možné používat vstupní data pro testy, což ale nakonec znamenalo lepší kontrolu dat a rychlejší průběh testů. Velkým přínosem tohoto projektu pro mne bylo užívání verzovacího nástroje Git<sup>1</sup>.

### 7.2 Budoucnost projektu

Budoucnost tohoto projektu je velmi otevřená. Aplikace pro hodnocení vín určitě existují, přednost tohoto projektu je řízení se oficiálními pravidly hodnocení vín organizace OIV. Možnosti rozšíření aplikace jsou široké. Je možné přidat prvky sociálních sítí a to od přidávání přátel, přidávání oblíbených vín, porovnávání svých hodnocení až po pořádání vlastních soutěží. Další možností rozšíření je implementace aplikace pro mobilní zařízení. Na soutěžích se předpokládá použití tabletů a aktuální verze pomocí prohlížeče nemusí být ideální. Také by bylo dobré uvažovat o přístupnosti aplikace. Tím myslím například možnost zvětšení písma a tlačítek, změny kontrastu a podobně. Nezkoušený uživatel by mohl mít problém se v aplikaci orientovat a proto by bylo dobré vytvořit tutoriál, který by uživatele naučil základní ovládání a hodnocení.

---

<sup>1</sup><https://gitlab.fel.cvut.cz/>





## Kapitola 8

### Závěr

Tato práce zahrnovala tři části: konfigurace soutěže, průběh soutěže, hodnocení veřejností. Povedlo se navrhnout a implementovat formulář pro konfiguraci soutěže. Ten organizátorům umožní měnit všechna nastavení soutěže na jednom místě. Průběh soutěže byl rozšířen o nový typ hodnocení pomocí hvězdiček. Proto byla aplikace zobecněna a budoucí přidání dalších typů hodnocení nezmění její logiku. Dále vznikla nová alternativa oficiálním soutěžím a to festival. Ten umožňuje veřejnosti hodnotit vína tak, jak je hodnotí komisaři na soutěžích. Veškerá funkcionalita byla řádně otestována pomocí automatizovaných testů.

Testování systému pomohlo odhalit chyby způsobené změnami struktury aplikace, ale také všeobecné programátorské chyby. Otestována byla veškerá funkčnost definovaná případy užití. To znamená, že byla otestována konfigurace soutěže. Dále pak průběh soutěže z pohledu tří uživatelských rolí a zároveň pro dvě různé soutěže, lišící se typem hodnocení (OIV, hvězdičky). Otestována byla také funkčnost festivalu.

Osobně si myslím, že největší uplatnění najde hodnocení hvězdičkami. Je to ideální způsob hodnocení na festivalu, snadno pochopitelný a porovnatelný. V budoucnu se určitě využije pro osobní hodnocení vín, kdy uživatelé si budou moci navzájem hodnotit vína. Tuto část oceňuji více také proto, že na rozdíl od podpůrných konfiguračních funkcí, je určena pro běžné uživatele této aplikace.



# Literatura

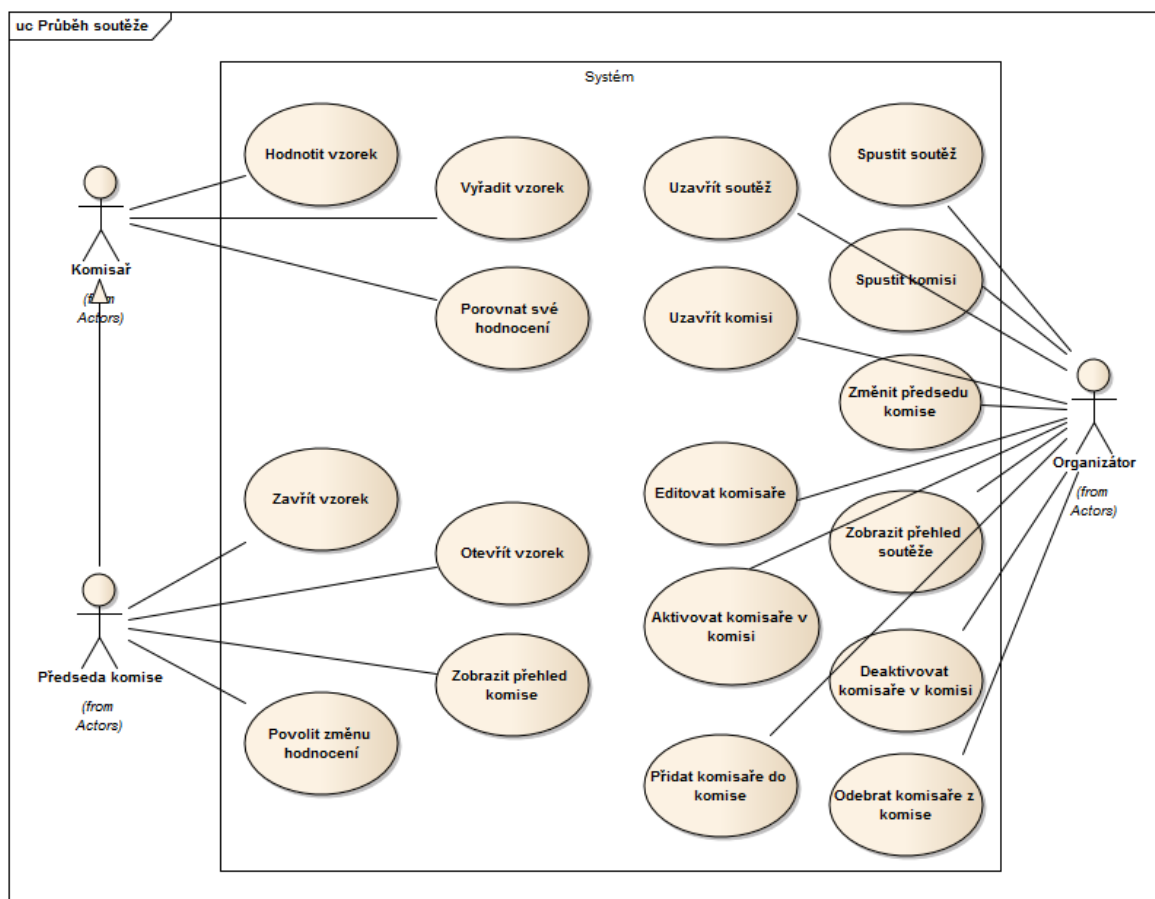
- [1] ARLOW, J. – NEUSTADT, I. *UML 2 a unifikovaný proces vývoje aplikací*. 1. nám. 28. dubna 45, Brno, ČR : Computer Press, 2007. ISBN 978-80-251-1503-9.
- [2] Bootstrap dokumentace. *JavaScript* [online]. 2015. [cit. 16. 5. 2015]. Dostupné z: <<http://getbootstrap.com/javascript/#popovers>>.
- [3] Django dokumentace. *Class based view* [online]. 2015. [cit. 23. 4. 2015]. Dostupné z: <<https://docs.djangoproject.com/en/1.7/topics/class-based-views/>>.
- [4] Django dokumentace. *Form and field validation* [online]. 2015. [cit. 15. 5. 2015]. Dostupné z: <<https://docs.djangoproject.com/en/1.7/ref/forms/validation/#cleaning-a-specific-field-attribute>>.
- [5] Django dokumentace. *Form fields* [online]. 2015. [cit. 15. 5. 2015]. Dostupné z: <<https://docs.djangoproject.com/en/1.7/ref/forms/fields/#filefield>>.
- [6] Django dokumentace. *Creating forms from models* [online]. 2015. [cit. 15. 5. 2015]. Dostupné z: <<https://docs.djangoproject.com/en/1.7/topics/forms/modelforms/>>.
- [7] Django dokumentace. *Django* [online]. 2015. [cit. 25. 4. 2015]. Dostupné z: <<https://www.djangoproject.com/>>.
- [8] Django dokumentace. *The Django template language* [online]. 2015. [cit. 15. 5. 2015]. Dostupné z: <<https://docs.djangoproject.com/en/1.7/topics/templates/>>.
- [9] Generální ředitel OIV. *RESOLUTION OIV/CONCOURS 332A/2009* [online]. 2009. [cit. 14. 4. 2015]. Dostupné z: <<http://www.oiv.int/oiv/info/enconours>>.
- [10] PECINOVSKÝ, R. *Návrhové vzory, 33 vzorových postupů pro objektové programování*. nám. 28. dubna 45, Brno, ČR : Computer Press, 2013. ISBN 978-80-251-1582-4.
- [11] Příspěvatelé Wikipedie. *Návrhový vzor* [online]. 2015. [cit. 16. 4. 2015]. Dostupné z: <[http://cs.wikipedia.org/wiki/N%C3%A1vrhov%C3%BD\\_vzor](http://cs.wikipedia.org/wiki/N%C3%A1vrhov%C3%BD_vzor)>.
- [12] Polymorphic dokumentace. *Welcome to django-polymorphic's documentation!* [online]. 2015. [cit. 14. 5. 2015]. Dostupné z: <<https://django-polymorphic.readthedocs.org/en/latest/>>.

- [13] Python dokumentace. *Python* [online]. 2015. [cit. 14.5.2015]. Dostupné z: <<https://www.python.org/>>.
- [14] Python dokumentace. *Built-in Functions* [online]. 2015. [cit. 15.5.2015]. Dostupné z: <<https://docs.python.org/3.4/library/functions.html#property>>.
- [15] Python dokumentace. *Unit testing framework* [online]. 2015. [cit. 15.5.2015]. Dostupné z: <<https://docs.python.org/3/library/unittest.html#module-unittest>>.
- [16] Selenium dokumentace. *What is Selenium?* [online]. 2015. [cit. 16.5.2015]. Dostupné z: <<http://www.seleniumhq.org/>>.

# Příloha A

## Modely případů užití

### A.1 Průběh soutěže



Obrázek A.1: Use case model - Průběh soutěže



# Příloha B

## Specifikace use case

### B.1 Konfigurace soutěže

- **UC 01 - Změnit typ hodnocení soutěže**

**Vstupní podmínky:** Žádný vzorek v soutěži ještě nebyl ohodnocen.

1. Případ užití začíná, když organizátor klikne na tlačítko „Nastavení“ u dané soutěže.
2. Systém zobrazí editační formulář, který obsahuje pole pro výběr loga, typu soutěže, maximální počet hvězdiček, způsob udělování medailí, absolutního vítěze a atributy vína. Aktuální hodnoty jsou předvyplněné.
3. Organizátor vybere typ hodnocení a odešle formulář.
4. Systém uloží nastavení soutěže a přesměruje na přehled soutěží.

- **UC 02 - Změnit logo soutěže**

1. Případ užití začíná, když organizátor klikne na tlačítko „Nastavení“ u dané soutěže.
2. Systém zobrazí editační formulář, který obsahuje pole pro výběr loga, typu soutěže, maximální počet hvězdiček, způsob udělování medailí, absolutního vítěze a atributy vína. Aktuální hodnoty jsou předvyplněné.
3. Organizátor vybere validní obrázek, který chce nahrát jako logo.
4. Systém uloží soubor a soutěž a přesměruje na přehled soutěží.

- **UC 03 - Vybrat viditelné atributy vína**

1. Případ užití začíná, když organizátor klikne na tlačítko „Nastavení“ u dané soutěže.
2. Systém zobrazí editační formulář, který obsahuje pole pro výběr loga, typu soutěže, maximální počet hvězdiček, způsob udělování medailí, absolutního vítěze a atributy vína. Aktuální hodnoty jsou předvyplněné.
3. Uživatel zaškrtně požadované atributy a odešle formulář.



4. Systém uloží nastavení soutěže a přesměruje na přehled soutěží.

• **UC 04 - Nastavit způsob udělování medailí**

1. Případ užití začíná, když organizátor klikne na tlačítko „Nastavení“ u dané soutěže.
2. Systém zobrazí editační formulář, který obsahuje pole pro výběr loga, typu soutěže, maximální počet hvězdiček, způsob udělování medailí, absolutního vítěze a atributy vína. Aktuální hodnoty jsou předvyplněné.
3. Organizátor vybere, jakým způsobem se mají udělovat medaile.
4. Systém uloží nastavení soutěže a přesměruje na přehled soutěží.

• **UC 05 - Nastavit počet hvězdiček pro hodnocení**

**Vstupní podmínky:** Žádný vzorek v soutěži ještě nebyl ohodnocen.

1. Případ užití začíná, když organizátor klikne na tlačítko „Nastavení“ u dané soutěže.
2. Systém zobrazí editační formulář, který obsahuje pole pro výběr loga, typu soutěže, maximální počet hvězdiček, způsob udělování medailí, absolutního vítěze a atributy vína. Aktuální hodnoty jsou předvyplněné.
3. Organizátor vybere počet hvězdiček a odešle formulář.
4. Systém uloží nastavení soutěže a přesměruje na přehled soutěží.

**Poznámka:** V průběhu implementace byl, pro přehlednost, editační formulář rozdělen do tří záložek (Obecné nastavení, Medaile, Zobrazení).

## B.2 Průběh soutěže

• **UC 06 - Hodnotit vzorek**

1. Případ užití začíná, když komisař klikne na tlačítko „Hodnotit“.
2. Systém zobrazí hodnotící formulář, ten obsahuje výpis viditelných atributů vzorku, komponenty potřebné pro typ hodnocení dané soutěže a také tlačítka „Potvrdit“, „Vyřadit“, „Porovnat“.
3. Dokud nejsou zadány všechny hodnoty:
  - 3.1 Systém zobrazí hlášku o nutnosti vyplnění všech hodnot.
4. Systém uloží hodnocení a zobrazí porovnání hodnocení.

Alternativní scénář:

**Hodnotit vzorek: částečné vyplnění**

1. Alternativní scénář začíná krokem 3.1 hlavního scénáře.
2. Komisař přeruší vyplňování hodnot a opustí formulář.

3. Systém uloží částečně vyplněné hodnocení.

- **UC 07 - Vyřadit vzorek**

1. Případ užití začíná, když aktuálně hodnocený vzorek nevyhovuje požadavkům.
2. Komisař klikne na tlačítko „Vyřadit“.
3. KDYŽ má komisař nastaveno zobrazování potvrzovacího okna.
  - 3.1 Systém zobrazí potvrzovací okno s možností vypnout toto potvrzování.
  - 3.2 Komisař potvrdí vyřazení vzorku.
4. Systém uloží komisařovo rozhodnutí o vyřazení vzorku.

Alternativní scénář:

**Vyřadit vzorek: nepotvrzení vyřazení**

1. Alternativní scénář začíná krokem 3.2 hlavního scénáře.
2. Komisař nepotvrdí vyřazení a pokračuje v hodnocení vzorku.

- **UC 08 - Porovnat své hodnocení**

1. Případ užití začíná, když komisař hodnotí vzorek.
2. Komisař klikne na tlačítko „Já“.
3. Systém zobrazí okno s přehledem komisařových hodnocení.
4. KDYŽ komisař klikne na tlačítko „Hodnotit“.
  - 4.1 Systém komisaře přesměruje zpět na hodnotící formulář.

- **UC 09 - Zavřít vzorek**

**Vstupní podmínky:** Vzorek je ohodnocen všemi komisaři v komisi.

1. Případ užití začíná na přehledu komise.
2. Předseda komise klikne na tlačítko „Potvrďte“.
3. Systém uzavře vzorek.

- **UC 10 - Otevřít vzorek**

**Vstupní podmínky:** Žádný vzorek není otevřen a existuje vzorek, který nebyl hodnocen.

1. Případ užití začíná na přehledu komise.
2. Předseda komise klikne na tlačítko „Otevřít vzorek“.
3. Systém otevře vzorek.

- **UC 11 - Zobrazit přehled komise**

1. Případ užití začíná, když předseda komise hodnotí vzorek.
2. Předseda komise klikne na tlačítko „Všichni“.

3. Systém zobrazí stránku s přehledem hodnocení aktuálního vzorku a s tabulkou všech vzorků a hodnocení v komisi.
- **UC 12 - Povolit změnu hodnocení**
    1. Příklad užití začíná na přehledu komise.
    2. Předseda komise klikne na tlačítko „Upravit“ u jména požadovaného komisaře.
    3. Systém umožní danému komisaři změnit své hodnocení.
  - **UC 13 - Spustit soutěž**
    1. Příklad užití začíná v okně s přehledem všech soutěží.
    2. Organizátor klikne na tlačítko „Spustit“ u dané soutěže.
    3. Systém spustí soutěž.
  - **UC 14 - Uzavřít soutěž**
    1. Příklad užití začíná v okně s přehledem všech soutěží.
    2. Organizátor klikne na tlačítko „Ukončit“ u dané soutěže.
    3. Systém ukončí soutěž, uzavře všechny komise a udělí medaile.
  - **UC 15 - Spustit komisi**
    1. Příklad užití začíná na přehledu soutěže.
    2. Organizátor klikne na tlačítko „Spustit“ u dané komise.
    3. Systém spustí komisi.
  - **UC 16 - Uzavřít komisi**
    1. Příklad užití začíná na přehledu soutěže.
    2. Organizátor klikne na tlačítko „Ukončit“ u dané komise.
    3. Systém uzavře komisi.
  - **UC 17 - Změnit předsedu komise**
    1. Příklad užití začíná na přehledu komise.
    2. Organizátor klikne na tlačítko „Upravit“.
    3. Systém zobrazí stránku s tabulkou komisařů v komisi.
    4. Organizátor klikne na přepínač ve sloupci „Předseda komise“ a v řádku daného komisaře.
    5. Systém změní předsedu komise.
  - **UC 18 - Editovat komisaře**
    1. Příklad užití začíná na přehledu komise.
    2. Organizátor klikne na tlačítko „Upravit“.

3. Systém zobrazí stránku s tabulkou komisařů v komisi.
4. Organizátor klikne na ikonu u jména daného komisaře.
5. Systém zobrazí formulář pro editaci komisaře s předvyplněným jménem a příjmením editovaného komisaře.
6. Organizátor vyplní položky Jméno, Příjmení, Heslo, Heslo znovu a odešle formulář.
7. Systém uloží změněné údaje komisaře.

Alternativní scénář:

#### **Editovat komisaře: hesla se neshodují**

1. Alternativní scénář začíná bodem 6. hlavního scénáře.
2. Systém informuje organizátora, že položky Heslo a Heslo znovu se neshodují.

#### • **UC 19 - Aktivovat komisaře v komisi**

1. Případ užití začíná na přehledu komise.
2. Organizátor klikne na tlačítko „Upravit“.
3. Systém zobrazí stránku s tabulkou komisařů v komisi.
4. Organizátor klikne na ikonu ve sloupci „Upravit“ a v řádku daného komisaře a potvrdí.
5. Systém aktivuje komisaře.

#### • **UC 20 - Deaktivovat komisaře v komisi**

1. Případ užití začíná na přehledu komise.
2. Organizátor klikne na tlačítko „Upravit“.
3. Systém zobrazí stránku s tabulkou komisařů v komisi.
4. Organizátor klikne na ikonu ve sloupci „Upravit“ a v řádku daného komisaře.
5. Systém zobrazí okno s výběrem akce.
6. Organizátor klikne na přepínač „Deaktivovat“ a potvrdí.
7. Systém deaktivuje komisaře.

#### • **UC 21 - Přidat komisaře do komise**

1. Případ užití začíná na přehledu komise.
2. Organizátor klikne na tlačítko „Upravit“.
3. Systém zobrazí stránku s tabulkou komisařů v komisi.
4. KDYŽ organizátor klikne na tlačítko „Přidat nového komisaře“.
  - 4.1 Systém zobrazí formulář pro vytvoření nového komisaře.
  - 4.2 Organizátor vyplní E-mail, Jméno, Příjmení, Heslo a Heslo znovu.
  - 4.3 Systém uloží nového uživatele.

5. KDYŽ organizátor klikne na tlačítko „Přidat komisaře z jiné komise“.
  - 5.1 Systém zobrazí formulář s výběrem komisařů.
  - 5.2 Organizátor vybere komisaře a potvrdí.
6. Systém přidá komisaře do komise.

Alternativní scénář:

**Přidat komisaře do komise: nelze přidat z jiné komise**

1. Alternativní scénář začíná v bodě 5.1 hlavního scénáře.
2. Systém upozorní organizátora, že nelze přidat komisaře z jiné komise.

**Přidat komisaře do komise: špatně vyplněný formulář**

1. Alternativní scénář začíná v bodě 4.2 hlavního scénáře.
2. Systém upozorní organizátora na chybně vyplněný formulář.

● **UC 22 - Odebrat komisaře z komise**

1. Případ užití začíná na přehledu komise.
2. Organizátor klikne na tlačítko „Upravit“.
3. Systém zobrazí stránku s tabulkou komisařů v komisi.
4. Organizátor klikne na ikonu ve sloupci „Upravit“ a v řádku daného komisaře.
5. Systém zobrazí okno s výběrem akce.
6. Organizátor klikne na přepínač „Smazat“ a potvrdí.
7. KDYŽ komisař nemá žádné hodnocení.
  - 7.1 Systém odebere komisaře z komise.
8. JINAK
  - 8.1 Systém deaktivuje komisaře.

● **UC 23 - Zobrazit přehled soutěže**

1. Případ užití začíná na přehledu soutěží.
2. Organizátor klikne na tlačítko „Vzorek“ u dané soutěže.
3. Systém zobrazí stránku s přehledem všech vzorků na soutěži a s průměry jejich hodnocení.

### B.3 Hodnocení veřejností

● **UC 24 - Zobrazit festival**

1. Případ užití začíná přesměrováním uživatele na úvodní stránku festivalu.
2. Systém zobrazí stránku s přehledem vzorků na festivalu.

● **UC 25 - Hodnotit vzorek**

1. Příklad užití začíná, když uživatel klikne na tlačítko „Hodnotit“ u vybraného vzorku.
2. Systém zobrazí hodnotící formulář, ten obsahuje výpis viditelných atributů vzorku, komponenty potřebné pro typ hodnocení daného festivalu a také tlačítko „Potvrdit“.
3. Dokud nejsou zadány všechny hodnoty:
  - 3.1 Systém zobrazí hlášku o nutnosti vyplnění všech hodnot.
4. Systém uloží hodnocení a zobrazí porovnání hodnocení.

Alternativní scénář:

**Hodnotit vzorek: částečné vyplnění**

1. Alternativní scénář začíná krokem 3.1 hlavního scénáře.
2. Uživatel přeruší vyplňování hodnot a opustí formulář.
3. Systém uloží částečně vyplněné hodnocení.

• **UC 26 - Změnit hodnocení vzorku**

1. Příklad užití začíná, když uživatel klikne na hodnocení vybraného vzorku.
2. Následuje scénář Hodnotit vzorek.

• **UC 27 - Porovnat hodnocení**

1. Příklad užití začíná, když uživatel klikne na popis vína u vybraného vzorku.
2. Systém zobrazí informace o víně a tabulku s porovnáním uživatelského hodnocení s hodnocením verifikovaných komisařů.
3. KDYŽ uživatel klikne na tlačítko pro hodnocení.
  - 3.1 Systém zobrazí hodnotící formulář.



# Příloha C

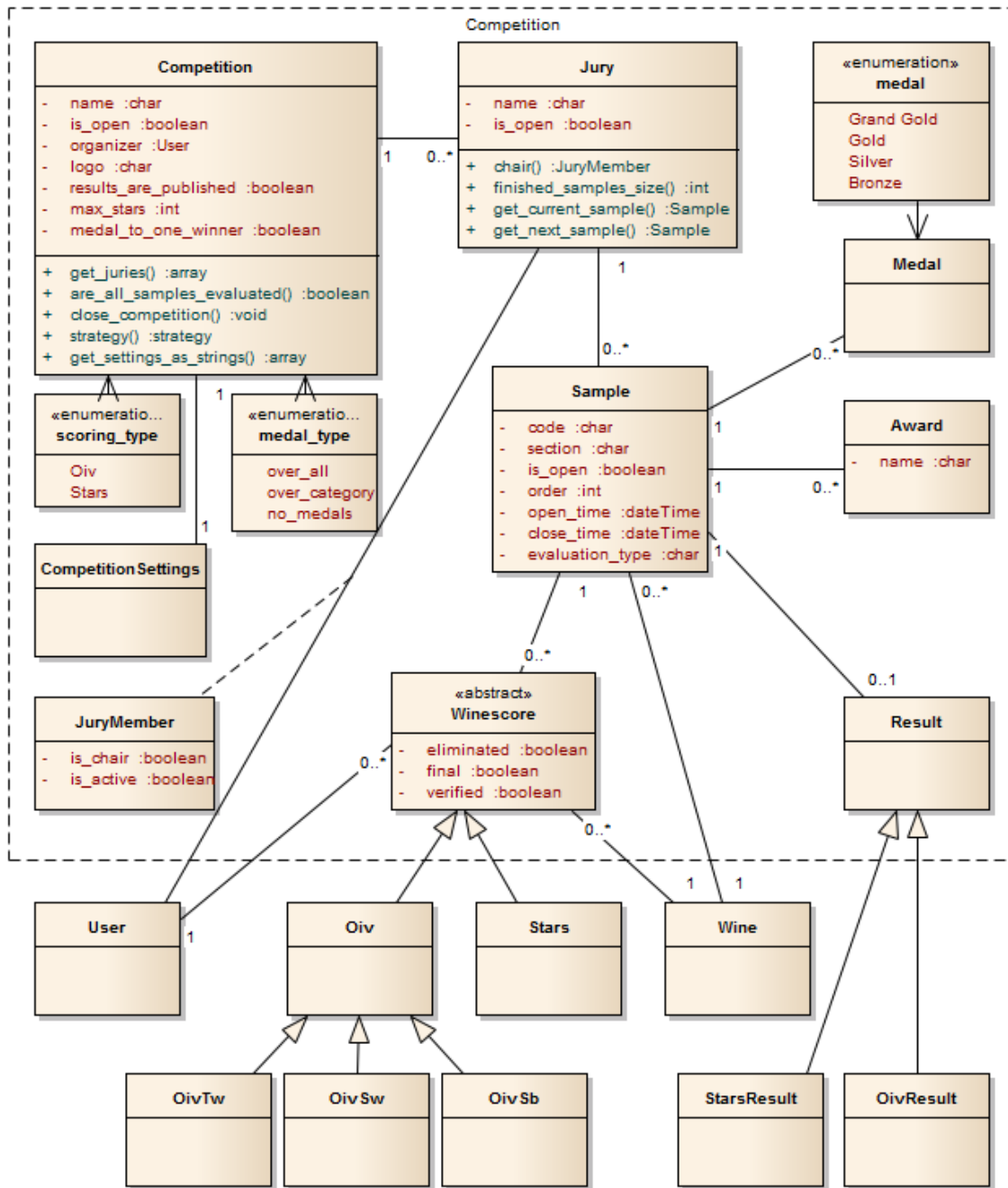
## Implementace

### C.1 Struktura projektu

```
project
├── Competition
│   ├── models.py
│   └── views.py
├── Jury
│   ├── templates
│   └── views.py
├── Oiv
│   ├── medals.py
│   ├── model_forms.py
│   ├── models.py
│   └── strategy.py
├── Organizer
│   ├── views.py
│   └── forms.py
├── Festival
│   └── views.py
└── Stars
    ├── model_forms.py
    ├── models.py
    └── strategy.py
```

### C.2 Model balíčku Competition



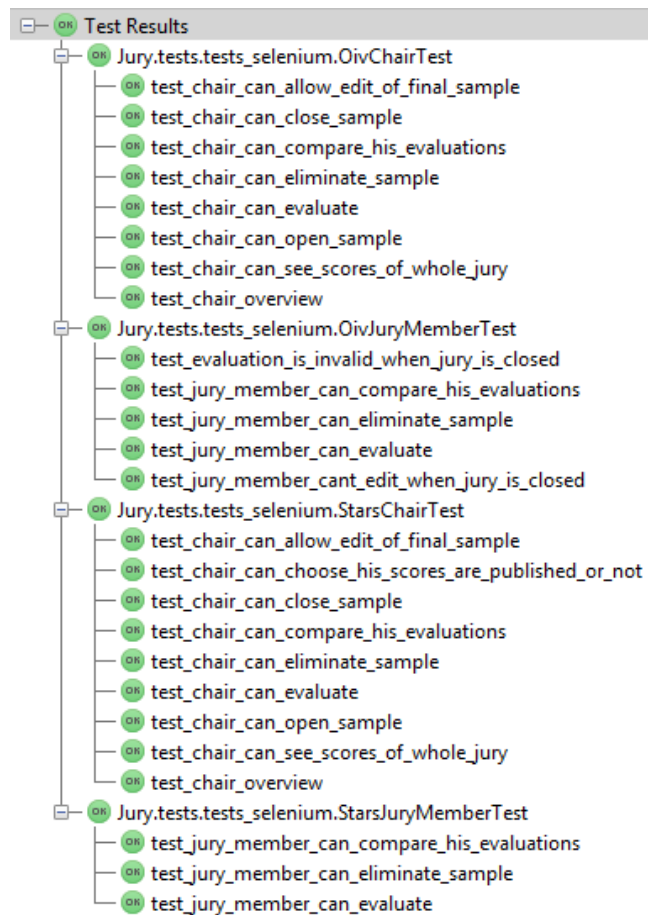


Obrázek C.1: Model tříd balíčku Competition

# Příloha D

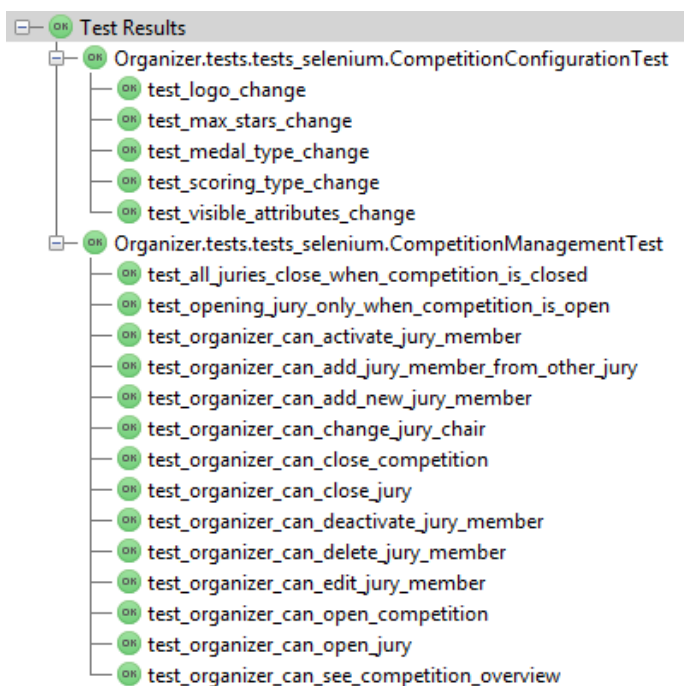
## Testování

### D.1 Jury



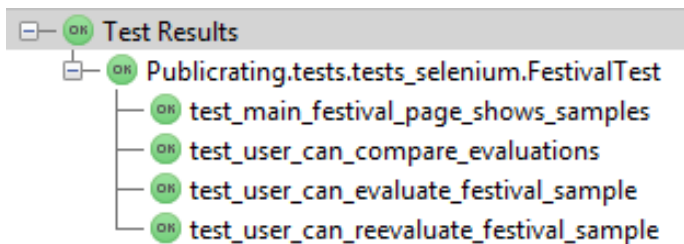
Obrázek D.1: Přehled testů balíčku „Jury“.

## D.2 Organizer



Obrázek D.2: Přehled testů balíčku „Organizer“.

## D.3 Festival



Obrázek D.3: Přehled testů balíčku „Festival“.

## Příloha E

# Obsah přiloženého CD

```
CD
├── Aplikace
│   ├── Oiv
│   │   ├── templates
│   │   ├── medals.py
│   │   ├── models.py
│   │   └── strategy.py
│   ├── Festival
│   │   ├── templates
│   │   └── views.py
│   ├── Stars
│   │   ├── templates
│   │   ├── models.py
│   │   └── strategy.py
│   └── tests
│       ├── Jury
│       │   ├── tests_selenium.py
│       │   └── tests_unit.py
│       ├── Organizer
│       │   └── tests_selenium.py
│       └── Festival
│           └── tests_selenium.py
└── text
    ├── Latex
    └── honsdomi.pdf
```